Micro Informatique

CPC • PCW • PC1512

CPC

REVUE DES STANDARDS AMSTRAD

TOUT SUR LES FICHIERS

REALISATION : UN AMPLI STEREO POUR VOTRE CPC



M 1355 28 20,00 F
3791355020009 00280

SOMMAIRE

EDITO

Nous avons une chance et un malheur.

Notre chance, c'est de ne pas avoir utilisé comme titre, AMSTRAD. Voilà qui nous évite certains procès et, dans tous les cas de figure, ce n'est pas notre politique. Depuis des années, nous prenons un soin particulier pour choisir nos titres, évitant autant que faire se peut, toute équivoque. Ainsi, marquons-nous notre indépendance !

Notre malheur se situe au niveau du contenu et... du titre.

Figurez-vous que, pour de nombreux lecteurs, nous sommes trop "pro". Si notre équipe prend cela comme un compliment, nous sommes dans l'obligation de reconsidérer notre politique du contenu. Nous savons que notre mensuel sert parfois de référence et qu'un rédacteur concurrent n'a pas hésité à faire un banc d'essai... en lisant sur le même matériel. Mais tout de même, de là à nous classer en "pro" !

Enfin, au niveau du titre CPC, certains lecteurs, nous pensons à tort, estiment que cette revue est dédiée aux CPC en général, à l'exclusion de tout autre. Or, ce n'est pas le cas et nous ne le voulons absolument pas!

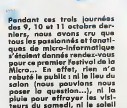
Enfin, on annonce pour début 88, la fin de CPC. Etant "le Patron", je devrais être au courant ! Alors, qui a intérêt à faire courir ce bruit ? Espérons que cette information, qui ne peut être "innocente", ne vient pas d'un fantôme hantant les vieux châteaux anglais !

S. FAUREZ
Directeur de la publication



VOYAGE AU COEUR DU FESTIVAL DE LA MICRO

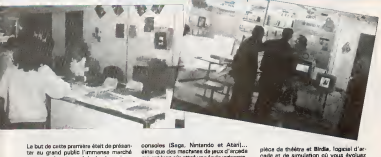




Pendant ces trois journées des 9, 10 et 11 octobre derniers, nous avons cru que tous les passionnés et fanatiques de micro-informatique s'étaient donnés rendez-vous pour ce premier Festival de la Micro... En effet, rien n'a rebuté le public : ni le lieu du salon (nous pouvions nous poser la question...), ni la pluie pour effrayer les visiteurs du samedi, ni le soleil pour dérouter les visiteurs du dimanche !







Le but de cette première était de présenter au grand public l'immense marché que représente celui de la micro-informatique. C'est pourquoi il était possible de découvrir des matériels de toutes marques, ainsi que des logiciels de jeux, éducatifs, de gestion ou de traitements de textes. Parmi les plus grands, il faut citer la présence d'Atari, de Thomson et de Commodore ; Amstrad était bien entendu absent, mais les passionnés, comme vous tous de CPC, ont pu se rassasier et toucher toutes les nouveautés grâce à la présence active de nombreux éditeurs quoique, là aussi, il y ait eu un grand absent en la personne de Loriciels, mais rassurez-vous, il nous concocte quelques nouveautés pour la fin de l'année... Pour terminer le tour d'horizon des participants à ce salon, il faut encore citer l'entrée en force des consoles (Sega, Nintendo et Atari)... ainsi que des machines de jeux d'arcade qui ont bien sûr attiré une foule indescriptible ; ce tableau ne pouvait bien sûr être complet sans mentionner la presse et plus particulièrement votre serviteur.

Du côté des nouveautés chez les éditeurs, nous avons vu notamment chez Ubi Soft **le Maître des Ames**, nouveau jeu de rôle dont vous trouverez un banc d'essai dans ce numéro ; pour Coktel Vision, c'est **Au nom de l'hermine**, aventure médiévale, qui était à l'honneur en attendant l'arrivée très prochaine de Blueberry et Astérix. (Ces trois logiciels sont également pour PC et compatibles). Ere Informatique, quant à lui, présentait deux produits tout juste terminés et dont vous trouverez très bientôt les bancs d'essais dans CPC ou Amstar : il s'agit d'**Oxphar** qui a déjà fait l'objet d'une pièce de théâtre et **Birdie**, logiciel d'arcade et de simulation où vous évoluez majestueusement sous la forme d'un oiseau blanc.

Pour ce qui est de la micro-musicale, Clavius, qui jusqu'à présent proposait des produits sur MSX, s'ouvre maintenant sur CPC avec notamment **Andrum** qui se révèle être une boîte à rythmes tout à fait honorable fonctionnant sur tous les CPC.

A noter pour les possesseurs de PC un logiciel remarquable de création graphique qui s'intitulant **Vectoria 3D** et dont l'auteur n'est autre que Marc Merzoug, fils de l'auteur de la Graphiscop...

Enfin, à titre informatif, sachez que la disparition de DK Tronics en Angleterre n'empêche pas la société Cameron en France de continuer à distribuer toutes les interfaces DK Tronics...

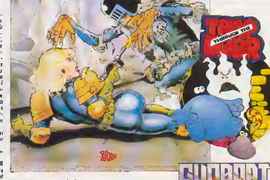
Pour terminer ce voyage, il nous faut parler de ce qui constituait l'événement du Festival de la Micro, j'ai nommé la grande finale des Jeux Micro-Olympiques... Cette compétition finale se jouait avec un logiciel jusqu'alors encore inédit : California Games, édité par Epyx. C'est un combat long et douloureux que les quatre finalistes ont dû livrer et ce, dans chaque discipline offerte par le logiciel : skateboard, surf, roller-skate, vélo cross, foot bag et le frisbee. Finalement, une fois que les jeux ont été faits, voici les gagnants (de gauche à droite sur la photo) :

- Médaille de bronze : Samuel YOUFI - 20 ans - Villejuif
- Médaille d'argent : David HACHOUR - 18 ans, Marly le Roi
- 4e finaliste : Laurent BEAUFILS - 18 ans - Mamers
- Médaille d'or : Michel HOUNG - 16 ans Bussy-St George



Le vainqueur de la finale est rentré chez lui avec un Amiga 2000 sous le bras ! Cette première édition du Festival de la Micro s'est déroulée pendant 3 jours avec une bonne participation du public, à quand la prochaine ?

• PIRANHA - Les bandes dessinées fournissent encore le thème d'un jeu informatique (mais où vont-ils chercher tout ça?) Le héros en question est YOGI BEAR, de la célèbre famille BEAR. Le thème est assez classique : Boo-Boo, le petit du groupe, a disparu. En tant que père dévoué, vous ne pouvez résister à l'appel du sang. Vous voulez partir dans le parc de Jellystone ? Faites vite! L'heure de l'hibernation approche! GUNBOAT vous met aux commandes d'un bateau ultra sophistiqué, rempli de torpilles dévastatrices. Vous avez la permission de casser tout ce qui bouge : ne vous privez pas ! D'autant que l'ennemi est aussi bien sur terre, dans l'air ou sur mer.
Le jeu suivant utilise un décor souterrain en toile de fond... THROUGH THE TRAP DOOR utilise les talents de 2 joueurs pouvant jouer alternativement Berk ou Drutt (véridique!) les héros partis au secours de Boni. JUDGE DEATH est un titre qui doit évoquer de tendres souvenirs aux passionnés de bandes dessinées américaines, puisqu'il s'agit de la version informatique de Judge Dredd. La décadence de Méga-City déplait assez aux Juges Noirs (les méchants). Ceux-ci considèrent que la vie même est un crime. Judge Anderson (la gentille, Rhââ lovely) va donc affronter Death, Martis, Fear et Fire en un combat plein de bruit et de fureur. Mr Weerms, comptable de son état, a décidé de devenir chasseur de vampires. Armé d'un fusil lance-ail et d'une réserve de cette liliacée odorante, le petit homme au complet gris s'engage dans l'entrée du sinistre manoir des vampires femelles... Va-t-il s'en sortir ?





• MARTECH - Les Anglais ne doutent vraiment de rien : pour eux, Nigel Mansell est le futur champion du monde de F 1 ! Martech tire profit de l'événement en annonçant la sortie, le 19 novembre, d'un logiciel à la gloire du pilote anglais. Ce programme est bien entendu une simulation de course automobile. Que les mégalomanes se rassurent, voici un logiciel pour eux : ARMAGEDDON MAN se propose de vous mettre à la tête d'un gigantesque réseau d'espionnage capable de diriger la politique de toutes les puissances nucléaires présentes sur terre. Si l'on ajoute un super satellite de défense et une tension certaine entre les 16 nouvelles super puissances, votre soif de pouvoir sera-t-elle satisfaite ?

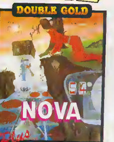
• DIGITAL INTEGRATION
Outre le "Going for gold" annoncé dans le précédent numéro, cette société spécialisée dans la simulation propose, pour une date indéterminée, un logiciel nommé ATF (Advanced Tactical Fighter). Inspiré d'un modèle réel, un Lockheed YF-22A cet ATF aura pour décor un paysage en 3D plein. La simulation ne concerne pas uniquement le vol puisque des combats sont prévus au menu.

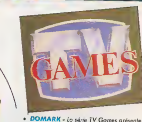


• ADDICTIVE - Les simulations de sport collectif n'ont pas encore, à de rares exceptions près, convaincu les foules. Pourtant Football Manager II est un logiciel de ce type, quelque peu remanié cependant. En effet, Manager ne signifie pas joueur : de lourdes responsabilités vont peser sur vos frêles épaules. C'est à vous qu'incombe le devoir de porter votre équipe à la victoire. Stratégie et tableaux d'arcade alternent dans ce jeu aux décors digitalisés (uniquement sur ATARI ST, hélas !)

• RAINBIRD
[illegible]

• INCENTIVE SOFTWARE
Sous le label "double gold" qui comprend principalement des jeux d'aventures, on trouve le logiciel Nova, plein de graphiques et de science-fiction. Cadeau gratuit : un deuxième jeu d'aventures Haunted House dont le nom veut bien dire ce qu'il semble être..





• DOMARK - La série TV Games présente Blockbusters, un jeu télévisé, assez populaire semble-t-il, sur l'île britannique. Le jeu mêle questions diverses et jeu de stratégie. Krypton factor n'est pas un jeu relatant les aventures de l'homme d'acier (autrement dit Superman), mais une reprise d'un autre "gameshow" nécessitant une réflexion rapide et un mouvement aisé du joystick.
Terminons la série TV Games avec un pack de 3 jeux : Countdown, Bullseye, Treasure Hunt. A rechercher pour l'exotisme.
Pas un sou de plus, pas un sou de moins, c'est ainsi que l'on pourrait traduire le titre de ce jeu d'aventure "new-look". Le scénario est original. Il s'agit de récupérer votre argent perdu dans une magouille organisée par une société concurrente de la vôtre. Vous comptez bien leur faire rendre gorge. Le jeu est rempli de graphismes et de voix digitalisées mais les puristes du texte seuls pourront supprimer toutes ces "distractions" visuelles et auditives.
Que la force soit avec toi ! Le premier épisode de la saga mondialement connue arrive sur CPC. Starwar vous décrit par le menu l'ultime bataille des rebelles contre la gigantesque Etoile de la Mort. De plus, les titres des logiciels suivants seront : The Empire Strikes Back et Return of the Jedi : les fanas de Georges Lucas vont pouvoir s'exprimer.



LORICIELS - Vous transporte en Norvège au XVII° siècle. C'est reparti pour de nouvelles aventures à la poursuite d'une cassette contenant les preuves de l'innocence de votre bien-aimée et de son père, emprisonnés injustement. Vous avez tout compris ? Bravo !
A propos, le scénario d'Han d'Islande a été écrit par un petit jeune : Victor Hugo. Cela vous dit quelque chose ?



BLANC MICHOT TELEMATIQUE - FW-Bureau est un logiciel écrit sous FREDII langage de programmation de FRAMEWORK II. Ce programme apporte à l'utilisateur 65 fonctions bureautiques dont la gestion multicritère du courrier construction automatique de lettres et documents, gestion d'un "répertoire" de noms et adresses. FW-Bureau fonctionne avec un IBM PC/XT/AT ou compatibles ayant un minimum de 512 K de RAM. Pour en savoir plus :
16 (1) 43 39 25 36.

LE SYNDROME AMSTRAD - Regardez bien la photographie du Spectrum +3. Cela ne vous rappelle rien ? Si vous répondez un ATARI 520 ST, vous avez perdu. Le petit dernier de Sinclair (alias Amstrad) ressemble en effet fortement au 6128. Le +3 reprend les mêmes caractéristiques techniques que son prédécesseur (le +2) avec, bien entendu, en plus, un lecteur de disquettes 3 pouces ce qui ravira vraisemblablement les possesseurs de modèles à cassettes (Remember les temps de chargement). On peut cependant se poser la question de la disponibilité des programmes compatibles Spectrum sur disquettes 3 pouces. Le prix : £199 en Angleterre.

SAVEZ-VOUS PLONKER ? - Vos disquettes sont éparpillées ou risque de subir une dégradation irrémédiable causée par un liquide contondant. Alors, adieu tableurs, traitements de textes. La Plonker 8ox se fixe sur le côté de votre moniteur ou de votre clavier. Cette boîte qui n'en est pas une, est percée de quatre fentes. Il suffit donc d'entrer un coin de la disquette 5 pouces dans un des artifices et le tour est joué. Toutes vos disquettes à portée de la main !

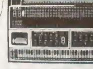
• **COBRA SOFT** : attention à vous en traversant le quartier Pigalle-Barbès car **LES RIPOUX** sont de sortie ! Le logiciel présente un cocktail maison d'actions, d'aventures avec un doigt de simulation qui risque fort d'être explosif.



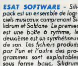
INTEGRALE DE PC EGALE - Il ne s'agit pas d'un nouveau logiciel mathématique sur PC, mais d'une offre faite par Amstrad à tout nouvel acheteur du PC 1512. Un logiciel "intégré" comprenant un traitement de textes (Evolution Sunset), un tableur graphique (Calcomat) et une base de données relationnelle (Superbase), vont s'ajouter à la configuration de base sans entraîner de modification de prix. Ces trois programmes sont liés par GEM et permettent donc un transfert avec des données d'un logiciel à l'autre.



INFOGRAMES - "Je veux être calife à la place du calife" Cette réplique sort tout droit de la bouche du (petit) grand vizir Iznogoud. Ce dernier, ainsi que son fidèle serviteur Dilatlarath, se sentait un peu à l'étroit dans une case de BD. C'est pourquoi vous pourrez bientôt les retrouver dans un logiciel d'arcade dont les décors et les personnages sont, bien sûr, ceux de la bande dessinée De plus, un album gratuit est fourni avec le programme, alors que les adorateurs de Tabary se précipitent sur leurs tirelires.







ESAT SOFTWARE - Silipack est un ensemble de logiciels musicaux comprenant Silidrum et Salitone Le premier est une boîte à rythme, le deuxième est un synthétiseur de son. Les fichiers produits par l'un et l'autre des programmes sont exploitables sous forme basic, Silidrum peut même utiliser les créations de Silitone et inversement, le tout étant géré par des menus déroulants

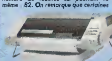
GO ! - C'est le nom d'une nouvelle branche d'US Gold ayant pour but la création de jeux aux graphismes et aux sons de qualité La naissance du petit a eu lieu en musique grâce au groupe de rock Resistor. C'est pourquoi vous trouverez sur la face B des cassettes GO ! une musique originale du groupe sus-nommé Retenez bien le nom suivant : Trantor (sous-titré the last Stormtrooper), les photos d'écran ont l'air superbes.





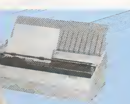
TROIS NOUVELLES IMPRIMANTES - Elles sont destinées à tous les compatibles et plus particulièrement aux PC 1512 et PC 1640.
La DMP 3160 ressemble comme deux gouttes d'eau aux DMP 2000 et 3000, même carrosserie et même système de chargement du papier. Les seules différences sont dues à la présence du jeu de caractères IBM ainsi qu'à des performances accrues : 160 CPS en mode texte et 40 CPS en "qualité presque courrier". Plus professionnelle semble être la DMP 4000 avec ses 132 colonnes et une vitesse de 200 caractères par seconde en mode normal ou 50 CPS en mode courrier.
Enfin la LQ3500, imprimante haut de gamme avec ses 24 aiguilles, vous donne une véritable qualité courrier (d'où le LQ : Letter Quality). Le papier est admis en feuille à feuille ou en continu grâce à la traction ou à la friction. Le bruit et la consommation électrique sont réduits. Ces caractéristiques en font une machine professionnelle, adaptable à tous les ordinateurs disposant d'une sortie parallèle (Centronics).

LE PCW 9512 - Après un PCW 8256 et un PCW 8512 on pouvait espérer un PCW 81024 Pas du tout, le prochain rejeton d'Alan Sugar est le PCW 9512. Ce nouveau système de traitement de textes possède un aspect comparable à ses prédécesseurs Le "design" a cependant été remanié et l'on note une certaine ressemblance avec un des membres de la nombreuse famille des compatibles.
L'outil se présente en trois éléments : le clavier, l'unité centrale plus le moniteur et enfin l'imprimante.
Le nouveau clavier apparaît plus large, plus aéré que celui des anciens PCW. Le nombre de touches est pourtant le même 82. On remarque que certaines



UNIWARE - Les amateurs de cartographie vont trouver chaussure à leur pied puisque ATLAS GRAPHICS est dédié à ce domaine particulier. Le logiciel est livré avec plusieurs cartes de France élaborées par l'IGN dont la réputation n'est plus à faire. Uniware vous propose ses cours de formation sur son produit ainsi que des applications "sur mesure". La configuration nécessaire est constituée par un compatible ayant au moins 512 Ko et si possible, un disque dur. Ce premier logiciel est interfacé avec une nouvelle version de Statgraphics (V 2.6). Comme son nom l'indique, Statgraphics est un programme d'analyses statistiques et graphiques, acceptant de nombreux périphériques de sortie, tels que imprimantes laser, traceurs etc. Atlas Graphics est proposé au prix de 3950 F (HT) et ce, jusqu'au 31.12.87. Pour plus de renseignements, contactez Uniware 16 (1) 45 27.20 61





touches sont affectées à des commandes du logiciel de traitement de textes Locoscript, d'où une simplification des manipulations.
L'écran haute résolution est en noir et blanc et couvre toujours une aussi grande surface, soit 90 colonnes et 32 lignes. L'unité centrale offre une mémoire vive de 512 Ko convertible si besoin en était en disque virtuel. L'unique lecteur de disquette encastré sur le côté gauche accepte des disques 3 pouces d'une capacité de 1 Mo non formatés. On trouve également l'emplacement pour un second disque de même capacité. On le voit, mis à part la capacité mémoire et le "look" de la carrosserie, il y a peu de différences entre l'ancienne et la nouvelle génération.
Le "plus" véritable du 9512 est à chercher du côté de l'imprimante. En effet, celle-ci utilise la technologie de la marguerite, c'est-à-dire que les caractères sont attachés sur une "corolle" pivotante qu'un mécanisme projette sur le ruban et imprime sur le papier. L'aspect de la machine est assez imposant. Souhaitons qu'elle ne mette pas trop en avant le défaut commun aux imprimantes à marguerite, c'est-à-dire le bruit.

On retrouve les options habituelles d'impression : caractères gras, soulignés, double frappe. En revanche, les changements de style de caractères obligent à une petite manipulation : il faut positionner une nouvelle "corolle" pour obtenir les styles Elite ou Courrier.
Les logiciels fournis avec la machine comprennent Locoscript II, le logiciel de traitement de textes auquel sont adjoints Locospell un correcteur orthographique et Locomail, un gestionnaire de "mailings". On trouve également le CP/M+ (et ses commandes graphiques GSX) ainsi que le célèbre BASIC MALLARD.
Amstrad ne vous laisse pas partir à l'aventure puisqu'il vous offre également un cours d'apprentissage sur cassette accompagné d'un livret illustré. De plus, une permanence téléphonique sera organisée pour vous guider en cas de problèmes.
Le PCW 9512 devrait être disponible à partir du 20 octobre au prix de 5490 F H.T.

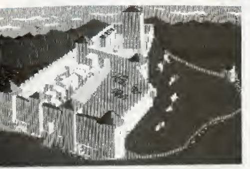
...erent characters available,
...opean Languages (as well as Greek)
of mathematical and currency symbols
availability of £ #, ß, f, $, ¤ and ¥)

This is in 17 pitch
is is in PS pitch
is in 12 (elite) pitch
in 10 (pica) pitch
in 17D (8½) pitch
in 12D (6) pitch
in 10D (5) pitch

• **ERE INFORMATIQUE :** ouvrez bien grand vos yeux et surveillez votre écran de CPC car vous risquez de voir très rapidement deux programmes très intéressants : d'une part, le grand projet d'**OXPHAR** est prêt... Rappelez-vous, ce spectacle fantastique a d'abord fait l'ojet d'une mise en scène théâtrale et maintenant, vous allez pouvoir vivre en direct cette épopée chevaleresque...
Parallèlement, les fanatiques de simulation vont pouvoir s'en donner à cœur joie avec un logiciel nouveau style s'intitulant **BIRDIE :** vous êtes un majestueux oiseau blanc et vous devez vivre une aventure fantastique en passant de mystérieuses portes qui vous propulsent dans des paysages parfois étranges, parfois inquiétants...

• **US GOLD :** dans la série adaptation de jeu de café, **OUTRUN** s'annonce comme une véritable course automobile grandeur nature avec une maîtrise remarquable de la 3D. Des promesses aussi alléchantes nous donnent fortement envie de découvrir la version sur Amstrad. Hélas, il va falloir attendre encore un mois !





• **COKTEL VISION :** étant donné le temps magnifique qui règne sur toute la France en ce moment, vous ne serez pas trop dépaysé en faisant un tour dans le grand Nord avec **GORBAF LE VIKING ;** pour parvenir à libérer son fils prisonnier d'un mage cruel, Gorbaf ne va cesser d'arpenter les caves humides et profondes...
Dans un tout autre ordre d'idée, **AU NOM DE L'HERMINE** vous entraîne dans une aventure médiévale en l'an de grâce 1249 ; ce logiciel entre dans la catégorie de programmes éducatifs à caractère ludique, vous permettant d'acquérir une culture générale sur le Moyen-Age...

COURRIER DES LECTEURS

ASSEMBLAGE

Jacky HINZE, des Vergers du Raquet voudrait en savoir plus sur le chargement des listings en langage d'assemblage du programme TRAMES ET COLLAGES de CPC n° 24

Tout d'abord, vous aurez noté que ce programme existe également sous la forme de DATAS et que vous pouvez le charger à partir du BASIC. Pour ce qui est des programmes en assembleur, vous devez vous procurer un programme appelé assembleur ou parfois moniteur, pour microprocesseur Z80, qui possède un éditeur qui vous permet d'entrer les mnemoniques (abréviations de commandes) et leurs attributs qui constituent le programme et qui seront traduits en code binaire par un interpréteur. Seul ce code est compris par la machine. Il vous faudra également vous munir d'un ouvrage d'apprentissage du langage d'assemblage Z80 si vous voulez aller plus loin que la saisie de programmes. Il existe sur le marché un certain nombre de ces programmes et ouvrages. Voyez votre revendeur.

MARGES

Guy ROUSSEL, de Gournay en Bray se demande comment initialiser son imprimante LX800 pour modifier les marges. Il voudrait décaler l'impression de ses programmes de manière à les insérer dans un dosseur sans que les perforations ne viennent amputer le texte

Nous ne savons pas précisément quels sont les codes de l'imprimante LX800 d'EPSON, mais la méthode est généralement celle-ci : supposons que le code d'échappement (ESCAPE) pour la marge gauche soit "1" et que le code pour la marge droite soit "Q". Supposons, d'autre part, que nous voulions placer la marge gauche à 20 colonnes du bord et la marge droite à 10 colonnes de la marge gauche - soit, sur une imprimante à 80 colonnes (PICA), à 70 colonnes de la marge gauche de référence - il nous faudra faire connaître notre intention à l'imprimante par la commande LPRINT, le code ESCAPE étant représenté par CHR$(27) comme à l'habitude. Voici les deux lignes à saisir :

```
10 LPRINT CHR$(27)+"1"
 +CHR$(20)   marge gauche
20 LPRINT CHR$(27)+"Q"
 +CHR$(70)   marge droite
```

Il va sans dire que ces marges resteront effectives jusqu'à réinitialisation de l'imprimante.

PCW/CPC

Est-il possible d'utiliser le BASIC MALLARD du PCW sur CPC et pour cela est-il nécessaire d'acheter un programme ?

Il est effectivement possible de programmer en MALLARD sur CPC. Les seules conditions à remplir sont, d'une part, de posséder la licence d'exploitation du BASIC en question - si l'on possède un PCW cela ne pose pas de problème de droit - et de posséder un CPC 6128, la mémoire utilisable après chargement sur les autres modèles n'étant que d'environ 9 Ko. L'opération se déroule sous CP/M 3.0 et il est nécessaire de reconfigurer le clavier. Vous saurez tout sur la procédure en lisant CPC n°17 à la page 60.

Disons tout de même que l'intérêt d'utiliser le BASIC MALLARD sur CPC est sa gestion de fichiers, notamment les fichiers indexés de JETSAM.

CAO

J'ai saisi le programme CAO sur CPC du n° 25 et je l'ai lancé sur mon CPC 464 et il ne tourne pas, nous dit Laurent CHARMES, de Clichy sous Bois

Il n'est pas étonnant que ce programme n'ait pas fonctionné sur votre 464 puisqu'il a été écrit pour 6128. Ce programme, comme indique en début d'article, fait au total 120 Ko - vous n'en avez saisi qu'un fragment -, la mémoire disponible du 464 étant de ....64 Ko. D'autre part, le programme utilise la fonction BANKMAN du 6128 qui donne accès à ses 64 KO de mémoire supplémentaire. En outre, si toutes les commandes du BASIC 464 se retrouvent dans le BASIC 6128, il existe un certain nombre de commandes de ce dernier modèle qui sont parfaitement inconnues du 464, à moins de créer ses propres RSX (CPC N°21 page 82). A ce propos, existe-t-il des lecteurs possédant une version gonflée du 464 - mémoire supplémentaire, DD1 - qui auraient utilisé BANKMANAGER POUR TOUS (CPC N°21 page 93) pour faire tourner CAO avec bonheur sur leur machine ainsi transfigurée ?

CABALISTIQUE

Jean FERARD de Cergy nous demande ce que peut bien être le signe se trouvant devant nn$ et an$ à la ligne 580 de SUPER DISC de CPC n°5

Ce signe étrange est le "@" que l'on appelle "arobas" ou "a commercial" ou encore "a rond". Vous le trouverez sur le clavier à la droite du "p".



PROTECTION

Afin d'optimiser le remplissage de mes disquettes, j'ai voulu recopier un programme de jeu que j'avais précédemment piraté et je l'ai perdu en cours de route. Pourriez-vous me dépanner ? Nous demande un lecteur dont nous tairons prudemment le nom

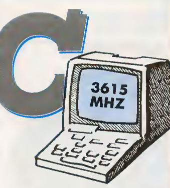
Croyez-vous vraiment, cher lecteur, que nous puissions participer, en toute candeur, à une entreprise de piratage ? N'oubliez pas que la loi punit, d'ailleurs plus sévèrement depuis peu, toute opération de piratage. Le respect de l'auteur et de la propriété de son œuvre ne peut pas être une vaine formule. Et si, vous-même, un beau jour, vous étiez un de ces auteurs piratés, tiendriez-vous le même discours ?



CPC SUR MINITEL

Vous le savez CPC vous offre une assistance technique par téléphone (A ce propos rappelons dates et horaires : mercredi de 9h à 12h et de 14h à 17h, ainsi que le vendredi matin de 9h à 12h). Mais hélas, submergés par notre succès, les deux lignes disponibles sont très souvent occupées. Tant et si bien que certaines personnes perdent tout espoir de voir un jour leurs problèmes résolus. Faut-il pour cela sombrer dans une noire mélancolie ? Pas du tout ! Il existe une autre voie : le minitel. Ce cube de plastique est un moyen de communication privilégié avec la rédaction puisqu'il n'y a plus de créneaux horaires. De plus la reponse est donnée généralement dans les 48 heures. Seulement la bonne utilisation du serveur n'est pas toujours évidente, c'est pourquoi pour tout ceux qui sont perdus dans les B.A.L, les annuaires et les pseudos, CPC vous propose un guide d'utilisation du serveur de CPC

Les opérations élémentaires sont connues de tous : le 3615 se compose sur le cadran du téléphone. Lorsqu'un sifflement retenti dans le combiné, il suffit de frapper la touche CONNEXION FIN pour obtenir la page de TELETEL 3. Le code d'accès est tout simplement MHZ (abrégé de Mégahertz). Une magnifique page de présentation (photo 1) éblouit alors vos yeux étonnés. On tape sur SOMMAIRE pour obtenir le menu "Choix des rubriques" (photo 2). Une série d'options, numérotées de 1 à 14 n'attend plus que votre bon vouloir. Pour le moment, puisque nous voulons écrire un message, seule l'option 2 va être décrite. Pour cela, tapez 2 suivi de ENVOI.
De nouveau, on trouve un menu en haut de l'écran. (photo 3). Première opération indispensable : la création d'une B.A.L. (ou boîte aux lettres). Tapez donc 1 plus ENVOI. A l'invitation du programme, entrez un nom de B.A.L., Astérix par exemple (photo 4) puis ENVOI. Ce nom



va nous permettre de vous identifier. Ensuite, il ne reste qu'à entrer un mot de passe (n'importe lequel pourvu que vous le mémorisiez facilement). En effet vous serez le seul à pouvoir lire le contenu de votre B.A.L. (photo 5), le programme vous demandera confirmation du mot de passe, puis le numéro de votre département ainsi que votre centre d'intérêt (ces deux dernières rubriques sont facultatives, elles permettent simplement une meilleure "signalisation" des correspondants).

De retour au menu, votre boîte aux lettres est enregistrée. N'oubliez pas que toutes les boîtes non utilisées durant 1 mois sont systématiquement effacées.

Passons maintenant à l'écriture d'un message. Imaginons que vous rencontriez un problème sur un listing de CPC, il va donc falloir nous envoyer vos doléances. La consultation de l'annuaire (option 3) vous permettra d'accéder à tous les noms des B.A.L (photo 6). Dans le cas de la rédaction de CPC, il faut choisir "CPC REDACT", autrement dit le numéro cinq. A ce propos, ne tapez pas directement le numéro de la B.A.L revenez plutôt au menu par sommaire et utilisez l'option numéro 5, c'est-à-dire ECRIRE. Une nouvelle page se présente (photo 7). Première question lettre à ? Tapez CPC REDACT, la date est affichée automatiquement. Deuxième question : de la part de ? Entrez alors votre nom de B.A.L (Astérix dans notre exemple), puis le sujet de votre message (facultatif). Vient l'heure de se servir de ses petits doigts pour entrer votre texte. Un dernier conseil : n'utilisez jamais ENVOI pour revenir à la ligne, sinon votre prose serait enregistrée telle quelle (Probablement sous forme incomplète). La touche ENVOI n'est donc utilisée qu'à la fin de la lettre. Voilà, votre message est enregistré, la réponse de CPC vous parviendra normalement sous 48 heures (photo 8). A bientôt.

APPRENDRE A PROGRAMMER EN BASIC AU CM2... POURQUOI PAS ?

J. REINGOT

A la suite de l'introduction de l'informatique à l'Ecole Primaire, j'ai été amené à conduire une expérience avec des élèves de Cours Moyen 2ᵉ année pendant ces deux dernières années. Je me propose de relater cette expérience en essayant d'analyser -sans prétention aucune- ses implications pédagogiques. Cela me paraît susceptible d'intéresser, outre les enseignants, les parents qui voudraient initier leurs enfants et au-delà tous ceux qui sont confrontés aux problèmes que pose l'apprentissage de la programmation. Je souhaite que les lecteurs intéressés participent à ce travail en faisant connaître leurs réflexions ou le résultat de leurs propres expériences.

Les enfants de 10 à 12 ans des classes de CM2 ont pour la plupart déjà eu des contacts avec un ordinateur, soit à la maison, soit en classe. Ils connaissent donc déjà les machines pour des activités ludiques ou éducatives sous la forme d'utilisation de logiciels.

L'expérience menée, au contraire, consiste dans l'apprentissage de la programmation en langage "BASIC" avec pour objectifs :
- l'acquisition d'un langage,
- le développement de l'esprit logique,
- l'aide psychologique au passage à l'abstraction,
- le développement de qualités de rigueur, d'ordre, de soin indispensables à ce genre d'activités et qui serviront évidemment dans d'autres disciplines.

Condition de déroulement de l'expérience

Matérielles : 7 machines (CPC 464 et 664) soit 2 ou au maximum 3 élèves par machine. Le travail s'effectuant par demi-classe à raison d'une séance de 1 heure et quart par semaine pour chaque élève.

Pédagogiques : le travail se fait en liaison constante avec l'institutrice de la classe ; c'est indispensable afin d'avoir une

utilisation réciproque des acquis en classe et en informatique, de pouvoir contrôler les connaissances et d'analyser les difficultés particulières rencontrées par tel ou tel élève.

Psychologiques : il est nécessaire de conserver l'aspect ludique de l'informatique tout en ayant le souci de "désacraliser" la machine. Le fait d'apprendre à programmer et d'être capable de faire exécuter à la machine ce qu'il a conçu apporte à l'enfant une nouvelle dimension dans ses rapports avec elle.

Chaque séance comportera si possible :
- une phase de tâtonnement,
- une phase d'expérimentation et de découverte,
- une phase d'apprentissage proprement dit,
- une phase d'invention et de réalisation.

J'introduis au fur et à mesure des besoins l'utilisation des instructions nécessaires : RUN PRINT EDIT LIST INK PEN BORDER etc
L'ordre dans lequel se font ces différentes acquisitions n'est pas strict.
Il dépend du programme suivi en classe, de la demande des élèves, de leurs progrès.
Certains chapitres s'étendent sur plusieurs séances, les différentes acquisitions sont réutilisées sans cesse.
Il s'ajoute à cela quelques séances consacrées à des jeux ; c'est la seule occasion de l'utilisation de logiciels.
Notons que (sauf pour le graphisme) nous utilisons du "BASIC STANDARD" et que l'emploi d'un autre matériel ne pose guère de problèmes.

Prise de contact avec la machine

Séance n° 1

Les élèves utilisent librement le clavier puis sont invités à expérimenter les différentes touches. Ils sont amenés à poser



des questions et on en arrive à expliquer l'utilisation des touches SHIFT, ESC, etc.
Il convient d'insister sur quelques points :
- utilisation de la touche ENTER (correspond à l'apprentissage de la leçon),
- différence entre le O et le 0,
- utilisation des flèches pour déplacer le curseur.
On arrive également à effectuer

Programme pour une année scolaire

NOTIONS ENSEIGNÉES	LANGAGE UTILISÉ	SUPPORT CONCRET
1 Maniement de la machine	CLS PRINT	
2 Qu'est-ce qu'un programme ? Notion de variable	GOTO	Arithmétique Numération
3 Structure Condition	IF THEN	" "
4 Boucle	FOR... TO NEXT	Arithmétique Numération - Tables
5 Entrées de données	INPUT	Calcul de surfaces Périmètres 4 opérations
6 Le "hasard"	RND INT	Nombres décimaux Partie entière, décimale
7 L'écran	LOCATE	Repérage dans le plan
8 Dessin Coordonnées d'un point	PLOT DRAW	Géométrie Parallèles, perpendiculaires Figures simples Déplacements
9 Chaînes Variables alphanumériques	RIGHT$LEFT$ MID$LEN	Orthographe Conjugaison Jeux avec des mots
10 Listes de données	READ DATA RESTORE SOUND	Dessins complexes Cartographie Musique



les 4 opérations en mode direct avec utilisation de PRINT et des signes opératoires + - = * / et point décimal
Les enfants aiment les couleurs, j'introduis l'instruction BORDER x ce qui permet de bien délimiter la fenêtre de travail sur l'écran. Je les laisse découvrir les limites du paramètre x.
Cette séance de manipulation de la machine doit suffire pour passer à la programmation lors de la séance suivante. Les élèves progresseront au fur et à mesure de leurs propres besoins dans la rapidité et la sûreté de frappe.

Qu'est-ce qu'un programme ? Notion de variable

Séance n° 2

On exécute une suite d'ordres élémentaires, par exemple :
lève-toi - prends un livre sur la table - pose-le sur le bureau - retourne à ta place - assieds-toi.
On numérote ces ordres : on a écrit un programme. On peut en inventer d'autres
Sur la machine on va taper un programme en prenant tout de suite l'habitude de numéroter de 10 en 10.

```
10  CLS
20  N=0
30  N=N+1
40  PRINT N
```

Les instructions CLS et PRINT étant connues, le seul gros problème c'est la ligne 30 bien que les élèves ne posent en général aucune question à son sujet. Il faut donc attirer leur attention sur ce qu'elle comporte de bizarre par rapport à l'arithmétique qu'ils utilisent régulièrement et leur expliquer qu'il ne s'agit pas d'une égalité mais d'une implication. Je leur fais utiliser le terme "devient" :
"n devient n+1 soit 0+1 c'est-à-dire 1"
"n qui était 0 devient 1"
Et on recommence en changeant 20 n=3
"n qui était 3 devient 3+1 c'est-à-dire 4"
ou en changeant 30 n=n+5
"n qui était 0 devient 0+5 c'est-à-dire 5"
et très rapidement les élèves peuvent deviner ce qui va s'inscrire sur l'écran chaque fois que l'on modifie les données des lignes 20 et 30 (une seule d'abord puis les 2 à la fois).
Il faudra revenir sans se lasser au cours des séances suivantes sur cette façon de procéder bien que j'aie constaté que les enfants assimilaient en général mieux que les adultes ces "variations" de la "variable".
Revenons au programme 1-1 et ajoutons "retourne à l'ordre n° 1".
Que se passe-t-il ?
Que se passe-t-il si on ajoute "prends un livre de plus que la fois précédente" ?
Ajoutons au programme 1-2 la ligne
50 GOTO 30
On tape RUN.. ENTER et

"Ah !!!"
Nous venons de créer le mouvement perpétuel..
Utilisons ESC une fois puis la barre..
Utilisons ESC deux fois puis la barre ? Peut-être avec RUN ?
On essaie d'expliquer ce qui se passe et de voir comment la "variable" n "varie". Il est nécessaire de faire répéter de nombreuses fois cette séquence.
On invente, on expérimente en modifiant le programme 2-2
Les expériences peuvent être variées :
- modifier la ligne 20 (initialisation)
- modifier la ligne 30 (incrémentation)
- modifier la ligne 50 en mettant GOTO 10 ou GOTO 20 (structure)
En fin de séance, chaque élève devrait être capable de réaliser un programme du type : "Compter de x en x à partir de y".
Cette séance est très riche : les enfants sont contents : ils font "marcher" la machine (et elle "marche vite" !): ils la font marcher "comme ils veulent" (et elle est obéissante). Ils commencent à assimiler la notion de VARIABLE essentielle en programmation ; ils ont fait un grand pas vers l'abstraction en remplaçant une donnée chiffrée par une lettre ; ils doivent travailler avec rigueur car ils se sont aperçus que la moindre faute de frappe ne pardonnait pas.
Remarques
1 - Il faudra revenir inlassablement sur le mécanisme de "variation" de la "variable".
2 - Les mots BASIC sont notés et devront être mémorisés ainsi que leur traduction en français :
Clear Screen (CLS) : Efface l'écran ,
RUN : Cours ! Mets toi en marche.
GOTO : Va à etc.
Les mots "savants" : initialisation, incrémentation, structure peuvent être appris. Ce qui est important c'est que les enfants en saisissent bien la signification et soient capables de répondre correctement à des consignes telles que : "Change la structure de ton programme" - "modifie l'incrémentation de la variable" etc.

LES ROUTINES MACHINE, BIEN SÛR ! MAIS OÙ ?



D'aucuns répondront immédiatement : "Dans une zone réservée par MEMORY évidemment !" Que voila une affirmation hasardeuse ! Bien entendu, notre position "anti-MEMORY" nous impose de présenter un procédé de "logement" des routines qui lui soit préférable dans bien des cas. Néanmoins, aucun procédé n'est la panacée. Il appartiendra toujours au programmeur de choisir entre les différentes méthodes possibles.



Bernard BESSE



La "Méthode" MEMORY EST-CE LA SEULE METHODE ?

Non, car on peut loger les routines aux endroits suivants

1 En dehors de la zone BASIC (adresses >HIMEM)

1.1 dans une zone fichier bidon

2 Dans la zone BASIC (de &0170 à HIMEM)

2 1 une ligne de remarque

2 2 une chaîne de caractères

2.3 une table d'entiers à une dimension

Ex DIM TOTO (100)

Notamment

La solution (1 1) paraît d'un emploi délicat en présence d'un fichier réel D'autre part, réserver un espace de 4096 octets pour y loger une routine de 100 octets ne paraît pas très judicieux

Les solutions (2 1) et (2 2) ont des limitations sérieuses, en particulier

(2 1) ne supporte pas de caractère &00

(2.2) ne peut loger que 255 caractères au plus

Nous n'étudierons donc ici que la solution 2 3

Logement dans des tables d'entiers à une dimension, une table DIM(N) pouvant contenir une routine de 2×N (et même 2×(N+1) octets

Mais d'abord, quels sont les inconvénients de MEMORY ?

Rappelons le processus de la méthode – le programmeur définit pour chaque routine une adresse telle que

a) la routine n'en recouvre aucune autre

b) et qu'elle ne s'étende pas au-delà de &AB7F (cassette) ou de &A67B (disque), ceci avec SYMBOL AFTER par défaut – une zone "ad hoc" est alors réservée par MEMORY

Si LOAD est la variable contenant l'adresse du début de la routine la plus basse, on fait alors

MEMORY ZLOAD-N' avec N le plus souvent égal à 1

Note : on peut, bien sûr, définir à priori un MEMORY (par ex 9FFF) et choisir l'adresse de la routine ensuite, mais c'est moins logique



Les inconvénients de MEMORY sont en conséquence les suivants

1) Si les routines sont installées dans une zone réservée surabondante, tout programme BASIC prenant de l'embonpoint risque un jour de se trouver à l'étroit Il faudra revoir alors le paramètre de MEMORY et probablement les adresses de chargement

2) Si la ou les routines sont installées à l'étroit, l'allongement d'une routine ou l'adjonction d'une routine supplémentaire conduira à des modifications analogues

Notons aussi que même dans le cas 1 (réservation surabondante) le programmeur devra s'assurer que les adresses de chargement n'entraînent pas de recouvrement

3) Rappelons enfin que toute réservation de mémoire (par MEMORY par exemple) pose le problème de la restauration de celle-ci. Voir CPC n° 23 "Restauration de la mémoire BASIC".

LA METHODE DES TABLES D'ENTIERS

Cette solution consiste à charger chaque routine dans une table d'entiers à une seule dimension

Elle ne s'applique qu'aux routines RELOGEABLES (c'est-à-dire qu'on peut char-

ger à une adresse "quelconque" voir CPC n° 19 "Déverouillons les RSX")
• Elle supprime l'utilisation de MEMORY, libère le programmeur du choix des adresses de chargement et supprime tout risque de recouvrement
• Enfin, après le dernier appel, toute routine peut être supprimée, libérant la place correspondante

Elle est particulièrement rentable pour les routines sous forme de OATA Mais on peut également transférer dans les tables d'entiers, les routines enregistrées sous forme de fichiers BINaires

Trois programmes sont joints au présent article
Le programme 1 montre simplement le logement (le chargement si l'on préfère) d'une routine de tri dans une table d'entiers
Le tri proposé fonctionne d'ailleurs parfaitement
Le programme 2 montre comment on peut transferer dans des tables d'entiers des routines stockées sous forme de fichiers BINaires
Le programme 3 propose aux superprudents une méthode pour s'assurer de la concordance Nbre DATA prévu <---> Nbre effectif

1 - Principe de la méthode "Tables d'entiers"

• Chargement à partir de DATA
Chaque DATUM est supposé être 1 octet sous forme HEXA
Supposons qu'une routine nommée CERCLE 8IN vienne d'être chargée dans la table ZMURF3, en commençant à l'élément ZMURF3(1)
Autrement dit, ZMURF3(1) contient les 2 premiers octets de la routine, ZMURF3(2) les octets 3 et 4, etc
Notons que ZMURF3(O) peut, si on le désire, être utilisé à la place de ZMURF3(1) comme début de routine (légères modifications à effectuer dans les programmes joints).
Pour passer le contrôle à la routine CERCLE, on écrit
CALL ∂ZMURF3(1),param1,param2
Mais ZMURF3() n'étant pas le nom le plus explicite, abandonnons-le et appelons CERCLE() la table où se trouve CERCLE (la routine)
L'appel devient
CALL ∂cercle(1), param1, param2
Ce qui est nettement plus clair
Pour être plus clair encore, on pourrait être tenté d'écrire (pour supprimer le ∂)
CERCLE = ∂cercle(1)
CALL cercle, param1, param2
Mais ce raffinement n'est pas sans inconvénients, comme nous le verrons

2 - Avantages de la méthode proposée

Rappelons-les :
— le 8ASIC gère lui-même les adresses des tables, donc celles des routines
— le programmeur n'a pas, non plus, à se préoccuper des risques de recouvrement (puisque le BASIC interdit celui des tables)
— l'ordre MEMORY disparaît ou devient d'une utilisation extrêmement réduite (ce qui est une bonne chose (voir CPC n° 23 déjà cité))
— la gestion dynamique des tables pour le 8ASIC garantit une perte de place faible entre 2 routines (15 octets environ) et permet également la récupération d'espace, comme dans l'exemple suivant

```
100 OIM cercle(100)
110'
120' ici chargement de la routine CERCLE de 200 octets dans la table cercle()
...
...
500 CALL ∂cercle(1), paramétres
                    Appels de CERCLE
...
990 CALL ∂cercle(1), paramètres
1000 ERASE cercle' on récupère la place prise par CERCLE()
```

3 - Inconvénients

En 8ASIC l'adresse d'une table peut varier, d'une part à cause des ERASE (les tables restantes serrent les rangs) et d'autre part, parce que toute nouvelle variable simple est insérée lors de sa création, avant les tables (à une adresse plus faible)
C'est ainsi que la création d'une variable entière ayant un nom de C caractères entraîne un decalage de 5 + C octets pour tous les tableaux déjà définis
Par conséquent, si l'on veut, pour être plus clair encore (comme on l'a vu) écrire
cercle = ∂cercle(1)
.... (rien ici)
CALL cercle, param1, param2
Il faudra éviter qu'une variable nouvelle vienne s'intercaler entre la saisie de l'adresse de la table (cercle = ∂cercle(1)) et le CALL lui-même
La meilleure solution est de supprimer tout risque en écrivant simplement, comme auparavant
CALL ∂cercle(1), param1, param2

Programmation

3 1 - Fichier DATA
On peut opèrer par exemple de la façon suivante
On lit d'abord n8 (qui a ici la valeur 46), nombre d'octets de la routine
La table "contenante" sera OIM (n16) avec n16 = n8/2 ou encore n16 = (n8-1)/2 + 1, ce qui évite de vérifier que n8 est pair (ou de le rendre pair par ajout d'un octet 00)

```
290 OATA 45
300 OATA DD,6E,04
310 DATA          C9

400 RESTORE 290 READ n8
n16=(nB-1)/2+1 : DIM toto(n18)
410 FOR k=1 TO n16
420 REAO 1$, m$
430 toto(k)=VALI("&"+m$+1$)'----
noter l'inversion m$,1$
440 NEXT k
```

3 2 Fichier 8INaira
Un MEMDRY provisoire est fixé, par exemple MEMORY &9FFF compatible avec la routine la plus longue qu'on peut envisager de charger dans le programme
Une adresse de chargement provisoire (prov = &A000 dans le programme joint) est fixée pour l'ensemble des routines à charger
Supposons qu'une routine TRI3 vienne d'être chargée à l'adresse PROV
La longueur N8 de la routine se trouve alors dans les octets &A760, &A76E (disque) et &8B1F, &8B20 (cassette)
Grâce à cette longueur N8, une table est créée de dimension
n16 = (n8-1)/2 + 1 d'où DiM tri3(n16)
La routine TRI3 est alors recopiée de l'adresse provisoire PROV à l'adresse "définitive" ∂TRI3(1)
Cette recopie est effectuée par une petite routine baptisée RECOP Cette routine qui est chargée la première doit se racopier elle-même dans la table RECDP()
3 3 - Quelques points à préciser
a) La routine RECOP a comme paramètres
• l'adresse du début de la zone SOURCE, qui est PROV
• l'adresse du début de la zone OESTINATION, qui est ∂RECOP(1) puis ∂TRI3(17, puis ∂TDTD(1) etc
• Le nombre N8 d'octets à racopier chaque fois
Voici la routine RECOP

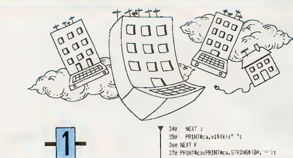
```
DATA D0,4E,00,00,46,01,DD,5E,02,
00,56,03,D0,6E,04,00
DATA 66,05,E0,80,C9,00
```

Nota cette routine n'est valable qu'en l'absence de recouvrement en zones "source" et "destination".
b) l'exécution du premier LDAD réserve un buffer de 4 K (&1000) octets
Pour les récupérer, il faut faire CLDSEIN à la fin du chargement des routinas
c) on peut alors récuperer la zone réservée initialemant par l'ordre du MEMORY du début

Dans le programme 2, nous utilisons à 2 endroits la procédure de restauration mémoire décrite dans CPC n° 23

Conclusion

4 1 - Routines en DATA
La simplicité et la souplesse du chargement dans des tables d'entiers est telle que l'ordre MEMDRY disparaît totalement de ces programmes (sauf pour les routines non relogeables et le MEMORY HIMEM-1 de l'OPENDUT "bidon")
4 2 - Fichiers BINaires
Nous considérons pour l'instant que le transfert des routines stockées en 8INaires dans des tables d'entiers ne constitue qu'un intéressant "exercice de style" qui ne prendrait toute sa valeur que si l'on pouvait se passer du chargement dans une zone provisoire et de RECOP Cette opinion serait moins défendable si le problème de la restauration de la mémoire BASIC était toujours non résolu Mais il l'est (voir CPC n° 23 déjà cité)



1

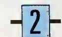
```
10   Programme 1 - Tri par insertion - Logement des DATA   >QB<
ds une Table
20                                                          >ZC<
21   (c) Bernard BESSE - 1987 -                             >BD<
22                                                          >BE<
30 WIDTH 80:CLS: DEFINT a-z: ca=8  (0 ou 8)                 >ND<
40                                                          >BE<
50   Nbre d octets de TRI12, routine                        >HF<
60 DATA 136                                                 >KG<
70 DATA DD,5E,04,DD,56,05,DD,6E,02,DD,66,03,DD,4E,00,DD     >RH<
80 DATA 46,01,C5,FD,E1,1B,B7,ED,52,E5,D5,FD,E5,FD,4E,00     >RJ<
90 DATA FD,6E,01,FD,66,02,D5,FD,46,FD,FD,5E,FE,FD,56,FF     >AK<
100 DATA C5,F5,D5,E5,04,0C,05,28,06,0D,20,08,B0,18,0D,0D    >QB<
110 DATA 78,B9,18,08,1A,BE,20,04,13,23,18,EA,E1,D1,C1,78    >FC<
120 DATA C1,38,19,28,17,FD,70,00,FD,73,01,FD,72,02,FD,2B    >DD<
130 DATA FD,2B,FD,2B,D1,1B,7A,B3,20,BC,18,01,D1,FD,71,00    >ME<
140 DATA FD,75,01,FD,74,02,FD,E1,FD,23,FD,23,FD,23,D1,13    >AF<
150 DATA E1,2B,7C,B5,20,93,C9,00                            >JB<
160                                                         >GH<
170  --------- Chargement de TRI12, routine ds TRI12, tab   >UJ<
le ---------
180                                                         >JK<
190 RESTORE 60:READ N8: N16=n8\2: DIM tri12(n16)            >VL<
200 FOR k=1 TO n16                                          >YC<
210   READ l$,m$                                            >LD<
220   tri12(k)=VAL("&"+m$+l$)                               >CE<
230 NEXT k                                                  >QF<
240                                                         >FG<
250  ------ Crea. nn chaines de long ll telle que  1<= ll   >VH<
 <= lg ------
260                                                         >HJ<
270 RANDOMIZE 13                                            >QK<
280 nn=50: lg=4: DIM v1$(nn), lt(lg)                        >XL<
290 FOR k=1 TO nn                                           >TM<
300   ll=INT(RND*lg)+1: lt(ll)=lt(ll)+1                     >XD<
310   v1$(k)=""                                             >GE<
320   FOR j=1 TO ll                                         >FF<
330     v1$(k)=v1$(k)+CHR$(65+INT(RND*26))                  >TB<
340   NEXT j                                                >RH<
350   PRINT#ca,v1$(k);" ";                                  >JJ<
360 NEXT k                                                  >VK<
370 PRINT#ca:PRINT#ca,STRING$(80,"-");                      >RL<
380                                                         >LM<
390  ------ CALL @tri12, deb(2 si tri complet), Dimension   >EN<
, @elem$(deb) ------
400                                                         >DE<
410 h1!=TIME                                                >AF<
420 deb=2:CALL @tri12(1),deb,nn,@v1$(deb)  ------ Appel d   >GG<
e TRI12
430 PRINT#ca:PRINT#ca,USING"#.##";(TIME-h1!)/300;:PRINT#c   >MH<
a," pour";nn;"chaines de longueur 1 a";lg
440 PRINT#ca                                                >DJ<
450 FOR k=1 TO nn:PRINT#ca,v1$(k);" ";:NEXT k               >NK<
460 PRINT#ca:PRINT#ca                                       >BL<
470 FOR k=1 TO lg:PRINT#ca,lt(k);"chaines de";k;"car. **    >BM<
";:NEXT k
480 PRINT#ca                                                >HN<
490 END                                                     >EP<
```

```
N E QPY BZD E JI MP EYR OEJ XY A LOPI V J EAC VCA PR MYCT L FTTB ES RBP E VTM RI
 IXJ DR YZC C JN JSY MP GTBZ PO JCY OER XXJ OY S WNE CI MU MKI W R ZHL VLQK G
JVCJ W
--------------------------------------------------------------------------------
0.00 pour 50 chaines de longueur 1 a 4

A BZD C CI OEJ DR E E E EAC ES EYR FTTB G GTBZ IXJ J JCY JI JN JSY JVCJ L LOPI
MP MU N MP OER OY PO PR QPY R RBP RI MKI S V VCA VLQK VTM W W WNE MYCT XXJ XY
YZC ZHL

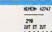
 14 chaines de 1 car. **  13 chaines de 2 car. **  17 chaines de 3 car. **  6
chaines de 4 car. **
```

Exemple de tri effectué à l'aide du programme 1.



```
10 ' Programme 2 - Logement de Routines stockees BIN dans  >Y8<
 des Tables
```

```
20                                                             >ZC<
21 ' (c) Bernard BESSE - 1987                                  >CD<
22 '                                                           >BE<
30 CLEAR:DEFINT a-z: ca=8  (0 ou 8)                            >PD<
40 GOSUB 490' ------ Restauration Memoire ------               >HE<
50 PRINT#ca,"HIMEM=";HIMEM                                    >AF<
60 PRINT#ca,"---------                                         >ZG<
70 '                                                           >EH<
80 '------ Chargement Routines ------                          >DJ<
90                                                             >GK<
100 prov=&A000:MEMORY prov-1                                   >HB<
110 '                                                          >BC<
120 LOAD"RECOP",prov                                           >KD<
130   GOSUB 560: DIM recop(n16)                                >RE<
140     CALL prov,prov,@recop(1),n8                            >RF<
150 LOAD"DEEK2",prov                                           >NG<
160   GOSUB 560: DIM deek2(n16): deek2=0                       >LH<
170     CALL @recop(1),prov,@deek2(1),n8                       >KJ<
180 LOAD"DOKE2",prov                                           >CK<
190   GOSUB 560: DIM doke2(n16)                                >JL<
200     CALL @recop(1),prov,@doke2(1),n8                       >PC<
210 LOAD"TRI12", prov                                          >TD<
220   GOSUB 560: DIM tri12(n16)                                >ZE<
230     CALL @recop(1),prov,@tri12(1),n8                       >PF<
240                                                            >FG<
250 ERASE recop:CLOSEIN:GOSUB 490'rest.MEM                     >PH<
260 '                                                          >HJ<
270 '------ TRAITEMENT------                                   >CK<
280 '                                                          >KL<
290 RESTORE 290                                                >HM<
300 CALL@deek2(1),&AE30,0,@deek2' Deek2 contiendra ADRESS      >GD<
E ligne
310 CALL@deek2(1),deek2,3,@deek2' Deek2 contiendra NUMERO      >TE<
 Ligne "Restoree"
320 PRINT#ca,deek2'  le No de ligne est affiche                >LF<
330 '                                                          >FG<
340 nom$="ZUT    ZUT"+"":CALL@deek2(1),@nom$,1,@deek2' AD      >GH<
R nom$ ds deek2
350 CALL@doke2(1),deek2,4,&5445' "ET" dokes avec decal.de      >UJ<
 6
360 PRINT#ca,nom$    et affichage nom$                         >ZK<
370                                                            >KL<
380 DATA S,X,R,T,A,I,J,O,P,W,Q,C,R,U,C                         >ZM<
390 DIM lettre$(15)                                            >ZN<
400 FOR k=1 TO 15:READ lettre$(k):NEXT k                       >CE<
410 CALL@tri12(1),2,15,@lettre$(2)                             >NF<
420 FOR k=1 TO 15:PRINT#ca,lettre$(k);" ";:NEXT k: PRINT#      >KG<
ca
430 '                                                          >GH<
440 PRINT#ca,"---------                                        >DJ<
450 PRINT#ca,"HIMEM=";HIMEM                                    >GK<
460 END                                                        >BL<
470 '****** SUBROUTINES ******                                 >BM<
480 '                                                          >KL<
490 :REM<R.MEM> ----- Restaur MEMOIRE BASIC au Maxi (SYM       >BP<
BOL after 256) ------
500 '                                                          >EF<
510 memlib'=PEEK(&AE7D)+256*PEEK(&AE7E):IF HIMEM<memlib'       >CG<
THEN MEMORY memlib':GOTO 510
520 hiold'=HIMEM: SYMBOL AFTER 256: IF HIMEM> hiold' THEN      >CH<
 510
530 maxi'=42747:IF HIMEM<maxi' THEN PRINT"HIMEM";HIMEM;"<      >AJ<
";maxi';" - Faire RESET
540 RETURN                                                     >NK<
550                                                            >KL<
560 :REM<N0> ------ N8 est le nbre d octets de la routine      >YM<
 ------
570                                                            >MN<
580 n8=PEEK(&A76D)+256*PEEK(&A76E): n16=(n8-1)\2 +1 -- Ad      >DF<
resses sys.Disque --
590 RETURN                                                     >UQ<
600 '#####@                                                    >JG<
```



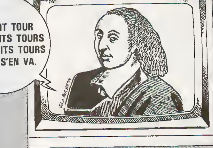
```
HIMEM= 42747
---------
 290
ZUT ET ZUT
A C C I J O P Q R R S T U W X
---------
HIMEM= 42747
```

Exemple d'exécution du programme 2.

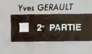
3

```
10   Programme 3 --- Controle Nbre DATA routine ---            >FB<
20 '                                                           >ZC<
21 ' (c) Bernard BESSE -1987 -                                 >BD<
22 '                                                           >BE<
30 DEFINT a-z: ON ERROR GOTO 210                               >TD<
40 ca=8' (0 ou 8)                                              >BE<
50                                                             >CF<
60 DATA 26                                                     >FG<
70 DATA DD,6E,04,DD,66,05,DD,5E,02,DD,56,03,19,5E,23,56        >JH<
80 DATA DD,6E,00,DD,66,01,73,23,72,C9                          >FJ<
90 DATA DEEK2                                                  >FK<
100                                                            >AB<
110 RESTORE 60                                                 >QC<
120 READ n8: n16=(n8-1)\2+1: DIM deek2(n16)                    >AD<
130 FOR k=1 TO n16:READ l$,m$:deek2(k)=VAL("&"+m$+l$):NEX      >HE<
T k:READ l$
140                                                            >EF<
150 RESTORE 90                                                 >YG<
160 READ n$:IF l$=n$ THEN 180                                  >YH<
170 PRINT#ca,"erreur nbre DATA de ";n$:END                     >WJ<
180 :                                                          >EK<
190 PRINT#ca,"nbres DATA concordent                            >ZL<
200 'mmmmmm                                                    >DC<
210 :REM<ERR>                                                  >GD<
220 IF ERR=4 THEN PRINT#ca,"Pas assez de DATA, ligne";ERL      >NE<
:END
230 ON ERROR GOTO 0                                            >DF<
240 '#####@                                                    >JG<
```

BRANCHER LE TURBO



Yves GERAULT



L'article du mois dernier nous donnais la règle du jeu pour l'utilisation de l'instruction CASE et le maniement des BOOLEENS. Voyons maintenant un exemple d'utilisation des BOOLEENS ainsi que deux nouvelles instructions IF THEN ELSE et GOTO.

APPLICATION DES BOOLEENS A L'INSTRUCTION CASE

Une limitation de l'utilisation de l'instruction CASE était que la variable de sélection ne pouvait être un réel, de plus les valeurs prises par cette variable devaient être fixées à la compilation et ne pouvaient être modifiées par la programmation. Voyons comment contourner ces obstacles

L'exemple choisi est la très classique équation du second degré $Ax^2 + Bx + C = 0$. Cela rappellera de bons souvenirs à certains et apprendra quelque chose à d'autres. Suivant les valeurs de A, B, C qui sont des réels, il existe une ou plusieurs solutions au problème. De plus ces solutions peuvent appartenir au domaine des réels ou à celui des complexes en fonction de la valeur de l'expression (B^2 4AC) appelée déterminant. Le détail des solutions à ce problème est donné au tableau 1 et la resolution en PASCAL à l'aide de l'instruction CASE est fournie par le listing 1

L'examen du listing nous montre la clarté de programmation apportée par l'instruction CASE pour un cas assez complexe. La première chose qu'il faut remarquer est la manière dont nous avons analysé le problème. Nous avons répertorié tous les cas de figure et nous leur avons fourni un numéro d'ordre en nous rappelant que l'ensemble des BOOLEENS était ordonné. La fonction ORD (vrai) donne la valeur 1 et celle ORD (faux) donne la valeur 0. Si nous avons correctement analysé le problème, seul un des BOOLEEN est vrai et tous les autres sont faux. Bien sûr si nous commettons une erreur dans cette analyse il risquerait sûrement de se passer des choses bizarres. Nous revenons sur un point important de l'utilisation d'un langage informatique quel qu'il soit : on doit passer plus de temps à analyser le problème qu'à le traduire en code compréhensible par la machine, en un mot ne pas confondre analyse et programmation.

L'INSTRUCTION IF..THEN..ELSE

Maintenant què nous savons pratiquement tout résoudre grâce à l'instruction CASE nous allons voir l'autre instruction de déroutement conditionnel IF THEN ELSE Si nous l étudions après il y a bien sûr une arrière pensée c est a cette instruction que le programmeur débutant fait le plus souvent appel lors de test, particulièrement lorsqu'il arrive du BASIC, car elle est très proche du langage parlé et nous souhaiterions que cette habitude se modifie au profit de l'instruction CASE plus structurée Il existe cependant des cas ou l'instruction CASE est vraiment trop lourde En particulier lorsqu'il s'agit de ne tester que le signe d'une variable par exemple Là, alors, c est réellement a l instruction IF THEN ELSE à laquelle il faut faire appel

Le principe de cette instruction repose sur la proposition suivante SI une proposition est vrai ALORS j'effectue une action SINON j'en effectue une autre Rien de nouveau par rapport au BASIC pensez vous ! Eh bien si et cela est dû à la structure en BLOC du PASCAL Comme dans le cas de BASIC il n'est possible d effectuer qu'UNE SEULE ACTION dans l'affirmative et qu'UNE SEULE ACTION dans le cas contraire, mais ces actions n'ont pas le même sens en PASCAL Une action en PASCAL consiste en une instruction, mais celle ci peut être simple ou multiple et là tout change Le tableau 2 donne la syntaxe de cette instruction dans ses trois utilisations possibles

La premiere consiste en la version simplifiée de cette instruction limitée à IF THEN A noter que cette instruction se termine par un point virgule (comme d'habitude !!, mais l'habitude est-elle prise ?).

- La seconde donne la syntaxe de l'instruction complète Il faut bien se souvenir de la différence entre la ligne EDITEUR et la ligne PASCAL pour PASCAL l'instruction commence avec le mot réservé " IF" et se finit avec le" ;" . Si pour des raisons de présentation il nous est plus agréable de mettre cette instruction sur 1, 2, .10, 100 lignes éditeur cela est admis par le compilateur mais cela ne change rien à la définition du début et à la fin de cette instruction

- La troisième nous indique la syntaxe à utiliser dans le cas général Il convient particulièrement de soigner la ponctuation, en particulier de ne pas mettre un" ; intempestif qui viendrait clore un peu hativement cette instruction Pour éviter des erreurs difficiles à détecter il est nécessaire dans cette instruction plus que dans toute autre d'adopter une présentation claire et soigner encore plus que d'habitude l'identation (alignement des BEGIN avec les END correspondants)

IMBRICATION DES INSTRUCTIONS CASE

Dans le cas de l'instruction CASE ce probleme est très simple à solutionner, cette instruction étant toujours "FERMEE" Par cela nous entendons que le début de cette instruction est parfaitement défini c'est le mot clef CASE, et la fin est aussi définie par le mot clef END Si de plus nous respectons scrupuleusement l'identation, ces deux mots devront être allignés et aucune confusion ne peut avoir lieu Bien sûr chacun des blocs INSTRUCTIONS MULTIPLES peuvent contenir des instructions CASE qui elles-mêmes... Il y a quand même une limite à ces imbrications successives, mais contrairement au BASIC cette limite n est pas numériquement définie, elle dèpend de l'espace mémoire disponible au moment de l'exécution (pile HEAP pour ceux qui sont BEAUCOUP plus avancés) De toute façon PASCAL vous le signalera par un message d'erreur lors de l'exécution

IMBRICATION DES INSTRUCTIONS IF..THEN..ELSE

Le probléme devient beaucoup plus ardu Cela est dû au fait que si le début de l'instruction est parfaitement défini par le mot clef " IF" il n'en est pas de même de la fin C'est d'ailleurs un des petits défauts du PASCAL (il faut quand même bien admettre qu'il n'est pas parfait, car alors comment expliquer l'existence des langages ADA et C) La fin de l'instruction est constituée par un point virgule, oui mais lequel ? Car dans le cas d'instructions multiples, il risque d'y en avoir pas mal De plus ceci se complique par le fait qu'il peut exister une version simplifiée (sans le ELSE) de cette instruction, dans ce cas à quel THEN le ELSE qui apparaît est-il rattaché ?
Il convient de réserver ces possibilités d'imbrication des IF THEN ELSE à des cas très particuliers et toujours se

Tableau 1 : Analyse du problème : recherche de solutions à l'équation : $Ax^2 + Bx + C = 0$.

On pose $D = B^2 - 4AC$

Cas 1 : A = 0 et B = 0 et C = 0
L'équation s'écrit 0 = 0. Quelque soit la valeur de x elle est vérifiée.
Il y a une infinité de solutions.

Cas 2 : A = 0 et B = 0 et C < > 0
L'équation s'écrit C = 0. Quelque soit la valeur de x elle n'est jamais vérifiée
Il n'y a aucune solution.

Cas 3 : A = 0 et B < > 0
L'équation s'écrit : Bx+C = 0. Elle n'est plus du deuxième degré mais du premier degré.
La solution est unique $x = -C / B$.

Cas 4 : A < > 0 et D = 0
L'équation est bien du second degré, elle admet deux solutions dont les valeurs sont égales.
Les solutions sont $x_1 = x_2 = -B / 2A$.

Cas 5 : A < > 0 et D > 0
L'équation est bien du second degré, elle admet deux solutions dont les valeurs sont différentes et appartiennent à l'ensemble des nombres réels.
Les solutions sont $x_1 = (-B + \text{sqrt}(D)) / 2A$.
$x_2 = (-B - \text{sqrt}(D)) / 2A$.

Cas 6 : A < > Ø et D < 0
L'équation est bien du second degré, elle admet deux solutions dont les valeurs sont différentes et appartiennent à l'ensemble des nombres complexes.
Les solutions sont $x_1 = (-B + i\,\text{sqrt}(D)) / 2A$.
$x_2 = (-B - i\,\text{sqrt}(D)) / 2A$.

Tableau II : Syntaxe de l'instruction IF..THEN..ELSE.

Première utilisation :

```
if expression booleenne then instruction ;
```

Deuxième utilisation :

```
if expression booleenne
 then instruction
 else instruction ;
```

Troisième utilisation :

```
if expression booleenne
 then
   begin
     instruction_1 ;
     instruction_2 ;

     instruction_avant_dernière ;
     instruction_dernière
   end
 else
   begin
     instruction_1 ;
     instruction_2 ;

     instruction_avant_dernière ;
     instruction_dernière
   end ;
```

demander s'il ne serait pas possible de transformer cette structure en CASE, ce qui simplifie les problèmes. Cependant s'il n'est pas possible de la faire ou simplement par jeu il convient de respecter les règles suivantes.

- Il ne doit jamais apparaître de " ; " juste devant un ELSE. Si vous venez de clore une instruction multiple qui se termine bien sûr par un END, celui-ci ne doit JAMAIS être suivi d'un " ; " si un ELSE suit.
- Ne jamais utiliser la version ebrégée du IF THEN ELSE quitte à créer des instructions vides si nécessaire (exemple ELSE ; Entre le mot clef ELSE et le " ; " il n'y a rien à exécuter). Ainsi vous saurez toujours à quel THEN se rattache le ELSE qui apparaît.
- Respecter impérativement les identations.

Le listing 2 donne un exemple d'utilisation de ces IF...THEN...ELSE imbriqués.

LES DEBRANCHEMENTS INCONDITIONNELS

Il s'agit du trop célèbre GOTO du BASIC. Cette instruction est tout juste tolérée en programmation structurée. Il est pratiquement toujours possible de la remplacer par une autre structure en PASCAL du fait de l'existence des BLOCS qui transforment une suite d'instructions simples en une instruction multiple. Cependant, puisqu'elle existe, il ne nous appartient pas de l'ignorer. Elle a quand même son utilité dans des cas bien précis : sortie d'urgence d'une structure PASCAL. Elle doit donc être utilisée avec très grande modération. Son usage conduit à des programmes illisibles et très difficilement modifiables. Nous avons preconisé le PASCAL pour justement éviter cette structure (entre autres choses), alors évitons la. Rappelons que PASCAL ne numérote pas ses lignes, le GOTO ne débranchera pas le programme vers un numéro de ligne c'est déjà cela de gagné. Il va donc être nécessaire de créer des reperes (des étiquettes) que nous allons placer a divers endroits du programme et auxquelles nous allons adresser le pointeur programme au gré de notre fantaisie. Le listing 3 nous donne une application de cette instruction.

Ces étiquettes (LABEL in english) sont constituées par un identificateur PASCAL, c'est a dire une suite de caractères (127 maxi) ne contenant aucun séparateur (voir articles précédents).

Pour pouvoir les reconnaître lors de la compilation, PASCAL impose qu'ils soient déclarés dans le BLOC DECLARATION dans un SOUS BLOC nommé LABEL. Comme le montre le listing 3, nous voyons que ce bloc est introduit par le mot clef " LABEL ". Les différentes étiquettes à déclarer sont ensuite énumérées suivant la manière habituelle en PASCAL : suite d'identificateurs séparés par des virgules. Il convient de noter que comme les SOUS-BLOCS CONST et VAR celui-ci ne contient aucun mot clef pour le terminer, seul le point virgule prévient le compilateur que la ligne PASCAL est terminée.

La syntaxe de l'instruction elle-même ne présente aucune particularite si ce n'est que le mot clef est GOTO et comme ce mot clef est un identificateur PASCAL il ne peut contenir de caractére separateur en particulier pas de " BLANC ". Petite différence avec BASIC.

L'instruction IF par contre montre un exemple nouveau de test. Nous désirons savoir si la variable CAR est contenue dans l'ensemble defini en énumération ['0'..'9'].

A suivre...

Listing 1

Listing 1 : Résolution de l'équation du second degré grâce à l'instruction CASE.

```
program equation_second_degre (input,output);
{=========================================}

var
      a,b,c,d,x1,x2            :  real;
      t1,t2,t3,t4,t5,t6        :  boolean;
      test                     :  integer;

begin
  clrscr;
  writeln ('Resolution des equations du second degre');writeln:
  writeln ('Entrer A B C separes par des blancs');
  readln (a,b,c);
  d:=b*b-4*a*c;
  t1:= (a=0) and (b=0) and (c=0);
  t2:= (a=0) and (b=0) and not (c=0);
  t3:= (a=0) and not (b=0);
  t4:= not (a=0) and (d=0);
  t5:= not (a=0) and (d>0);
  t6:= not (a=0) and (d<0);
  test:= 1*ord(t1)+2*ord(t2)+3*ord(t3)+4*ord(t4)+5*ord(t5)+6*ord(t6);
  case test of
        1    : writeln (' il y a une infinite de racines');
        2    : writeln (' il n''y a aucune racine ');
        3    : writeln (' il y a une seule racine : x = ',-c/b:9:5);
        4    : writeln (' il ya deux racines egales : x = ',-b/(2*a):9:5);
        5    : begin
                writeln (' il y a deux racines relles ');
                writeln ('x1 = ',(-b+sqrt (d))/(2*a) :9:5);
                writeln ('x2 = ',(-b-sqrt (d))/(2*a) :9:5);
               end;
        6    : begin
                writeln (' il y a deux racines complexes');
                writeln ('x1 = ',(-b/(2*a)):9:5,' + ',
                                                     sqrt (-d)/(2*a):9:5,'i');
                writeln ('x1 = ',(-b/(2*a)):9:5,' - ',
                                                     sqrt (-d)/(2*a):9:5,'i');
               end     {ATTENTION PAS DE ; AVANT LE ELSE }
      else     writeln ('il y a une erreur quelque part');
  end
end.
```

Listing 2

Listing 2 : Résolution de l'équation du second degré grâce à l'instruction IF..THEN..ELSE.

```
program equation_second_degre (input,output);
{=========================================}

var
      a,b,c,d,x1,x2            :  real;

begin
  clrscr;
  writeln ('Resolution des equations du second degre');writeln;
  writeln ('Entrer A B C separes par des blancs');
  readln (a,b,c);
```

```
                d:=b*b-4*a*c;
                if a =0
                  then if b = 0
                          then if c = 0
               then writeln (' il y a une infinite de racines')
               else writeln (' il n''y a aucune racine ')
          else writeln (' il y a une seule racine : x = ',-c/b:9:5)
    else if d = 0
          then writeln (' il ya deux racines egales : x = ',-b/(2*a):9:5)
          else if d > 0
               then
                begin
                  writeln (' il y a deux racines relles '):
                  writeln ('x1 = ',(-b+sqrt (d))/(2*a) :9:5):
                  writeln ('x2 = ',(-b-sqrt (d))/(2*a) :9:5)
                end
               else
                begin
                  writeln (' il y a deux racines complexes');
                  writeln ('x1 = ',(-b/(2*a)):9:5,' + ',
                                      sqrt (-d)/(2*a):9:5,'i'):
                  writeln ('x1 = ',(-b/(2*a)):9:5,' - ',
                                      sqrt (-d)/(2*a):9:5,'i')
                end
end.
```

Listing 3

Listing 3 : Exemple d'utilisation de l'instruction GOTO.

```
program exemple_goto (input,output);
{==================================}

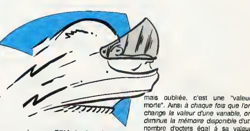
label
      etiquette_i,etiquette_2,etiquette_3;

var
      car    : char:

begin
  etiquette_1:
  write ('entrer un caractere : '):readln (car):
  if car in ['0'..'9'] then goto etiquette_2 else goto etiquette_3;
  {instruction}
  {instruction}
  {instruction}
  etiquette_2:
  writeln ('ce caractere est un chiffre ,on recommence');
  goto etiquette_1:
  etiquette_3:
  writeln ('ce caractere n''est pas un chiffre ,c''est fini'):
end.
```

COMMENT GAGNER PLUS, EN OCTETS ET EN VITESSE

Michel ARCHAMBAULT



Ou comment moins gaspiller en programmation BASIC. Cette suite de méthodes, d'astuces et d'exemples permettent d'améliorer considérablement "l'aérodynamisme" d'un programme et, comme en automobile, on obtient alors des améliorations importantes de la vitesse et de la consommation (ici en octets).

La taille du programme et sa vitesse d'exécution peuvent paraître comme choses mesquines pour des listings relativement courts, mais cela devient brutalement angoissant lorsque l'on aboutit à une grosse application. Alors autant prendre tout de suite quelques bonnes habitudes systématiques.

Par chance toutes les "radineries" en octets améliorent la vitesse, et d'autre part les astuces purement "turbo" n'augmentent jamais la taille du listing. Mais attention ! Il y a un risque, un seul, mais de taille : la CLARTE du listing. Il faut toujours se dire "Est-ce que dans quelques mois je comprendrai mon astuce d'aujourd'hui ?". Il faut faire la part des choses, et ce n'est pas toujours facile : faire une subtile économie de 70 octets, et l'expliquer dans un REM de deux lignes (80 octets) serait stupide.

LA SATURATION MEMOIRE ou "GARBAGE".

La grave erreur est de se croire à l'aise avec un programme de 15 kilo octets alors que la mémoire disponible d'un CPC est de l'ordre de 40 K. Dès que le logiciel "tourne" il crée des variables qu'il doit loger dans la mémoire restante. Normal, mais beaucoup de gens ignorent que *les valeurs périmées restent en place* dans la RAM. Exemple, suite à un INPUT, on a P$ = "PAUL" ; cela va se mettre en deux endroits : la zone des valeurs où PAUL occupe ses quatre octets, et la zone des "pointeurs de variables" où l'on lit que P$ est une chaîne de longueur 4 et que sa valeur se trouve à partir de telle adresse mémoire. A présent P$ = "ALBERT" ; ces 6 octets nouveaux sont mémorisés *ailleurs*, à une autre adresse référencée dans le pointeur de P$. Donc PAUL occupe toujours sa place de 4 octets en RAM, mais à une adresse désormais oubliée, c'est une "valeur morte". Ainsi *à chaque fois que l'on change la valeur d'une variable, on diminue la mémoire disponible* d'un nombre d'octets égal à sa valeur précédente. Tous les BASIC présentent ce défaut.

On devine alors qu'au bout d'un certain temps d'utilisation, il n'y aura plus de place libre et c'est le plantage temporaire appelé "GARBAGE". Cela dure au moins un quart d'heure pendant lequel le clavier semble inerte car l'ordinateur vide le "cimetière des valeurs mortes" en remontant les valeurs actuelles, celles qui avaient une adresse de pointage. Ce long travail terminé, le programme repart tout seul à la ligne où il s'était bloqué ; et ce jusqu'au prochain garbage.

On peut provoquer cette "vidange" par PRINT FRE(""), mais c'est aussi long, sauf s'il y a très peu de valeurs à traiter (RUN vide à la fois les valeurs actuelles et périmées mais instantanément) ; ne pas confondre avec FRE(0) qui renvoie *sans attente* la taille de la mémoire encore disponible. Ce petit test est assez "parlant"

```
10 PRINT FRE(0)
20 FOR N = 1 TO 5000 N$ = STR$(N) NEXT
30 PRINT FRE(0)
```

La mémoire disponible chute ainsi

de 42182 à 18273
Comment éviter une telle mésaventure ?
1 – En ne surchargeant pas la mémoire par des valeurs trop longues, trop nombreuses et fréquemment renouvelées
2 – En laissant davantage d'octets disponibles grâce à un programme BASIC moins encombrant en RAM

ECONOMIE SUR LES REM

La pire des choses est de faire de "superbes" (?) titres dans des cadres tout en astérisques, comme s'il s'agissait de la couverture d'un livre. Ah que c'est beau en tête de listing ! Faisons la facture : 10 lignes de 40 caractère = 400 octets (les blancs comptent) + 5 lignes de telles floritures pour chaque début de sous-programme = 200 octets. S'il y a dix sous-programmes, nous en sommes donc à 2400 octets gaspillés en RAM (et 3K sur Disc), pour que ce soit joli pour papier...
Le plus drôle de l'histoire est que ce rite nous vient des "vieux professionnels" ! En effet, cette pratique est recommandée pour taper les "programmes sources" dans les langages à compiler, tels le COBOL, FORTRAN, PASCAL, C, etc.. Cela fait de bons points de repères dans de très longs listings et sans la moindre conséquence, puisque les REM sont ignorés par les compilateurs, donc *absents dans le programme final*.
Or, il faut savoir qu'ils restent présents dans les langages interprétés (= non compilés), tels le BASIC, LOGO, DBASE II et quelques autres.
Certes, il nous faut des REM, mais ils seront succincts et courts. Un listing ce n'est pas fait pour être mis sous verre ou en vitrine (ou alors prévenez-moi !). Pour repérer très vite certaines lignes, il n'existe encore rien de plus efficace que les MARQUEURS FLUO : exemple rose pour les titres de sous-programmes, jaune pour les passages importants en milieu de ligne. Nous ne pouvons pas colorier ainsi les listings de notre revue CPC, c'est à vous de le faire.
LES REM HORS PROGRAMME pour de gros programmes BASIC, il est très prudent (pour soi-même) de noter une foule d'explications, en particulier la signification de toutes les variables utilisées. Sur une feuille de papier vous risquez de l'égarer, alors plutôt que de taper cela dans le listing, ouvrez tout simplement un FICHIER ASCII sur la même disquette. Exemple au programme MACHIN.BAS correspondra le fichier MACHIN.REM.
Deux méthodes pour écrire ce fichier : le petit programme ECRIFICH (il fait 10 lignes courtes) publié dans CPC n° 26 page 37, ou l'écrire sous BASIC après avoir tapé AUTO et l'enregistrer par SAVE "MACHIN.REM", A , inutile de commencer vos lignes de texte par une apostrophe, il n'y aura pas de Syntax Error...
Pour récupérer ce document taper ICPM puis TYPE MACHIN.REM, suivi ou non par CTRL P pour l'imprimer.
Ainsi le programme et ses explications sont indépendants (c'est souvent plus pratique), et sans supplément de travail.

LES ECRANS DE TEXTES

Vous connaissez la question "Voulez-vous les instructions ? (O/N)". C'est très louable que d'intégrer le tégrer la mode d'emploi sur la disquette, mais occuper chaque fois la mémoire par de nombreux kilo-octets qui ne servent qu'à la *première utilisation* du logiciel ne paraît pas très rationnel !
Là encore, nous pouvons avoir recours à un fichier ASCII MACHIN.DOC, donc extérieur, qui s'affichera à l'écran par vingt lignes à la fois. Cette routine (par OPENIN) est courte, elle a été publiée dans CPC n° 18 page 24, avec en prime un petit éditeur pour taper ce fichier.
Une lourdeur fréquemment rencontrée est l'affichage d'une liste de mots ou phrases par autant de LOCATE et de PRINT. Non seulement c'est une corvée à écrire mais c'est encombrant. Exemple : 18 chaînes à afficher à partir de la ligne 2 et en commençant colonne 10
– Les 18 chaînes sont mises dans une ou deux lignes de DATA (entre guillemets s'il s'agit de phrases)

```
FOR N = 1 TO 18 : READ
A$:LOCATE 10,N+1:PRINT
A$:NEXT
```

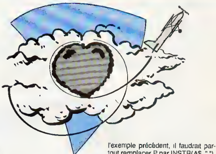
N'est-ce pas plus élégant que 18 lignes de LOCATE PRINT ? Et ce sera plus commode pour les modifications (ajout ou suppression de chaînes). Au lieu de la tabulation en 10, on peut alors en profiter pour s'offrir des centrages automatiques par

```
LOCATE (40–LEN(A$))
/2,N+1:PRINT A$
```

Lorsque l'on est en face d'un travail de tâcheron, il faut *réfléchir* afin d'obtenir la même chose mais avec le minimum d'efforts. On est gagnant sur tous les tableaux, du moins en programmation.
LES ECRANS-TITRES de présentation soignée sont agréables à regarder, sauf quand ils mettent des dizaines de secondes à se construire ! Sur cassette, on ne peut faire autrement, mais sur disc, il est préférable de charger un fichier écran de 17 K. Supposons que la

pour des réels et trois pour des chaînes Exemple . DIM V(200, 4) fait 201 x 5 = 1005 cases ou cellules, donc 5025 octets

Vous constatez qu'il ne faut pas être trop large en établissant ses DIM, la facture monte vite ! Dans un tableau de calculs, l'erreur "gaspi" est de prévoir une colonne pour y loger des résultats obtenus à partir des colonnes de données Mieux vaut refaire ce calcul chaque fois que l'on a besoin du résultat



ASTUCES AVEC LES FONCTIONS CHAINES

En mode 2, on veut afficher un trait avec 80 signes "moins" (-, code ASCII 45), alors le débutant programme un PRINT " suivi de 80 tirets, = 82 octets avec les guillemets ! Alors que PRINT STRING$ (80, 45) fait la même chose avec seulement TROIS octets (le TOKEN + 2 paramètres) Sans commentaire

Une autre fonction ultra pratique est INSTR, en voici deux exemples

– Vous devez rechercher la présence d'un mot M$, entré par INPUT, dans une colonne d'un tableau DIM NOM$(300) · la condition IF M$ = NOM$(N) est une vérification très stricte (le signe égal), elle est bien plus lente que IF INSTR (NOM$(N), M$)

En précisant IF INSTR(...) = 1 on s'assure que M$ n'a pas été rencontré au milieu du mot en DIM Par cette méthode, on peut se contenter d'entrer en M$ seulement les premières lettres du mot à chercher

– Un menu propose six réponses possibles A B C D E O
R = INSTR("ABCDO", R$)
Si R = 0 la réponse R$ est refusée Si elle est admise, la valeur de R permet d'enchaîner sur ON R GOTO (ou GOSUB) 2000, 3000, etc Sans le recours à INSTR, il aurait fallu programmer six IF au lieu d'un.

NOTA pour valider une réponde qui aurait pu être entrée en minuscules, on voit encore des IF R$ = "N" OR R$ = "n" THEN . De grâce, utilisez UPPER$ par R$ = UPPER$ (R$)

MID$ est bien connu mais on ignore très souvent l'effet intéressant obtenu en omettant le paramètre longueur

A$ = "ABCDEF" · PRINT MID$(A$, 4) donne "DEF", = ce qu'il reste à droite Cela nous dispense souvent de concevoir des formules complexes (donc lentes) du style RIGHT$(A$, LEN(A$)-X)

Exemple 1, insérer "****" dans A$ avant le premier D rencontré
P = INSTR(A$, "D") IF P>0 THEN A$ = LEFT$(A$, P-1) + "****" + MID$(A$, P)

Exemple 2, supprimer tous les blancs dans A$
120 P = INSTR(A$, " ") IF P = 0 THEN 140
130 A$ = LEFT$(A$, P-1) + MID$(A$, P+1) GOTO 120
140 PRINT A$

LES VARIABLES INTERMEDIAIRES

Dans les calculs ou manipulations complexes, on a tendance à créer des variables correspondant à des résultats intermédiaires Le listing paraît certes plus clair, mais on consomme de la mémoire pour des variables dont on n'aura plus besoin deux lignes plus loin Le remède consiste alors à remplacer cette variable par sa formule , ainsi dans l'exemple précédent, il faudrait partout remplacer P par INSTR(A$, " "), quitte à risquer de se perdre dans les parenthèses imbriquées

Rassurez-vous, on n'opère ainsi que lorsque le danger est réel, à savoir un passage au cœur d'une boucle, parce que très répétitif, surtout s'il s'agit de chaînes pouvant être longues

La plus grosse gaffe, hélas classique parce que tentante, consiste à concevoir une saisie de chaînes au clavier à l'aide de INKEY$ concaténés Exemple (à ne pas suivre) qui oblige à saisir un nombre déterminé de caractères
A$ = "" WHILE LEN(A$) < 15
A$ = A$ + INKEY$
WEND

Supposons que l'on tape la phrase "ordinateurs CPC" La variable chaîne A$ va prendre successivement les valeurs suivantes o, or, ord, ordi, ordin, etc On va consommer ainsi 120 octets au lieu de 15 avec un INPUT banal Donc cette seule entrée va faire grossir le cimetière des valeurs mortes de 105 octets ! Et en portant cette longueur maxi à 255 caractères, on arriverait à 32640 octets bonjour le garbage

LA LENTEUR DES IF

C'est la fonction courante la plus coûteuse en millisecondes Il faut s'arranger pour que le déroulement du programme en rencontre le moins possible, et c'est facile, la preuve

routine BASIC pour faire ce dessin fasse 3 K, c'est courant, faisons-en un *programme à part* se terminant par SAVE "TITRE1.ECR", B, &C000, &4000

Dans le programme principal, exempté de ces 3 K, on programme seulement CLS LOAD "TITRE 1.ECR". Et ça ne prend que cinq secondes, bien moins encore si vous savez exploiter les possibilités de votre 6128 : dans le "BANK" vous pouvez tenir en RAM trois images écrans *permutables* quasi instantanément avec celle actuellement affichée.

Cette technique des fichiers extérieurs (textes ou écrans) est courante dans les logiciels du commerce mais elle est volontairement rare dans les programmes publiés dans CPC, car cela compliquerait singulièrement la compréhension des lecteurs débutants lors de la saisie des listings. En revanche, reconnaissez qu'il vous sera facile désormais de les tronquer en fichiers extérieurs.

Les possesseurs de 664 ou 6128 ont la possibilité de fractionner un long listing en plusieurs sous-programmes rechargeables par CHAIN MERGE. Voir CPC n° 10 page 25 (ou "PROGRAMMES UTILITAIRES pour AMSTRAD" chapitre X)



LONGUEURS DE LIGNES ET DE VARIABLES

La ligne BASIC que vous tapez sera à votre insu pré-codée lors du SAVE. C'est aussi cette forme codée qui sera présente en RAM après un LOAD. On y trouve le numéro de ligne (1 octet), le numéro de la ligne suivante, et bien sûr le contenu. Chaque mot BASIC n'est représenté que par son "TOKEN", c'est un numéro de référence sur un octet, même LOCATE suivi d'un espace n'occupe ici qu'un seul octet au lieu de sept. En revanche les REM et les chaînes entre guillemets sont reproduits en clair. Vous comprenez alors pourquoi un programme enregistré en ASCII par SAVE"xxx", A tient plus de place sur disquette ou cassette que sa version normale en xxx.BAS.

Déduction évidente, plutôt que d'écrire une instruction par ligne, on a intérêt à mettre plusieurs "clauses" sur une même ligne, séparées par des doubles points (:). On aura ainsi moins de numéros de lignes, d'où une économie en octets et aussi une (légère) amélioration de la vitesse de transition d'une clause à une autre.

Evitez les noms de variables trop longs du genre

INITIALE$ = "M". N'est-il pas choquant d'utiliser ici un nom sur huit octets pour désigner une valeur "M" de un octet seulement ? On croit être plus clair avec des noms non abrégés, c'est vrai pour la première rencontre, mais si cette variable est fréquemment utilisée dans le listing vous remarquerez que c'est franchement très pénible à lire. (Cela rappelle les noms complets des personnages dans le théâtre russe classique !)

Dans la mesure du possible, mettez le maximum de vos variables avec des "noms indicés", c'est-à-dire dans des tableaux DIM, le gain en mémoire est appréciable.

LES VARIABLES NUMERIQUES

Super important ! Utilisez le plus possible les nombres entiers au lieu des "nombres réels" pris par défaut par le CPC. C'est facile puisque la plupart des valeurs numériques utilisées dans un programme sont non-décimales et comprises entre -32767 et +32767. Exemple N% = 23458 occupe deux octets alors que N = 1 en prend cinq.

Le gain en vitesse est spectaculaire : ainsi la boucle FOR N = 1 TO 10000 NEXT prend 11 secondes avec N en réel, mais 5, 5 secondes avec N%, soit exactement la moitié !

NOTA : nous écrivons NEXT au lieu de NEXT N. Le BASIC n'a pas besoin que vous lui signaliez à quelle variable numérique il doit attribuer le NEXT qu'il rencontre, ça le retarde de lui demander cette vérification, d'où une durée augmentée de 5 % environ.

Les boucles FOR NEXT ne sont pas les seules gagnantes, les calculs simples du style K = K + 1 au sein d'une boucle ont une vitesse augmentée de 30 % avec les variables entières. A utiliser impérativement dans toutes les routines de tris, surprise garantie ! Rappelons au passage que la commande DEFINT A–Z (ou I–N) placée en début de listing nous dispense de faire suivre chaque nom de variable entière par %.

Certains se demandent comment chronométrer le BASIC : avant le passage à tester, on programme T0 = TIME (ou T01 en cas de DEFINT A–Z), après le passage T1 = TIME puis PRINT (T1–T0)/300, "sec".

LA RESERVATION DES DIM

Une déclaration de tableau DIM a pour conséquence immédiate une réservation d'octets *même s'il est encore vide*. Comptez par "case" deux octets pour des entiers, cinq

```
100 INPUT "Quelle est la marque de votre voiture ?", V$
110 IF V$ = "RENAULT" THEN GOTO 400
120 IF V$ = "PEUGEOT" THEN GOTO 400
130 IF V$ = "CITROEN" THEN: GOTO 400
160 IF V$ = "OPEL" THEN GOTO 400
370 IF V$ = "CADILLAC" THEN : GOTO 400
380 IF V$ = "BENTLEY" THEN GOTO 400
390 PRINT"Marque mal orthographiée": GOTO 100
400 ' suite du programme
```

Vous remarquerez que les marques sont testées dans un **ordre de probabilités décroissantes** (Qui a dit hélas ?) avec **sortie immédiate** vers la fin des tests. Ainsi le programme subira en *moyenne* moins de sept IF au lieu de 28 systématiques.

Méfiez-vous des IF situés dans des boucles, surtout les FOR NEXT imbriquées, tel ce piège à secondes :

```
FOR N = 1 TO 20
FOR J = 1 TO 30
GOSUB 2000
NEXT:NEXT
```

Si le module en 2000 comporte un IF le programme va rencontrer 20 x 30 = 600 IF ! Vous vous souvenez que FOR N% = 1 TO 10000:NEXT ne dure que 5, 5 secondes ; insérons dans la boucle une condition pourtant jamais vérifiée, IF A$ = "ZZZZZ" THEN B$ = "XXXX" : la durée passe alors à 25 secondes.

En revanche un IF supplémentaire bien placé peut accélérer un traitement par "court circuit" :

```
500 FOR J = 1 TO 200
510 IF M$(J) = "" THEN 600
520 IF M$(J) = "FIN" THEN J = 200:GOTO 600
530 ' traitement complexe et long de M$(J)
..............
..............
600 NEXT
```



LE TRI INDEXE

Supposons un tableau DIM FICH$(250, 7) que l'on veut trier alphabétiquement sur telle rubrique-colonne. La pire bêtise est de lui faire subir un tri "physique" en permutant ses contenus de lignes avec leurs huit rubriques ; car même avec la méthode de tri la plus rapide, l'opération peut durer près de **quarante minutes !**

Il ne faut surtout pas toucher à ce gros tableau DIM, mais créer à côté un TABLEAU D'INDICES, du genre DIM S%(200), qui en fin de tri va contenir la bonne suite des numéros de lignes du gros tableau. Au départ ce DIM S% est garni de 1 à 200, et au cours du tri ce sont ces nombres entiers qui vont être déplacés. La durée chute alors de quarante minutes à **25 secondes !**

Pour avoir en RAM le gros DIM trié, on opère ensuite en deux temps : enregistrement du fichier en appelant ses fiches selon l'ordre de tableau d'indices. Lecture-chargement de ce fichier dans DIM FICH$. Durée totale tri indexé + enregistrement + rechargement = 40 secondes au lieu de quarante minutes.

Nota : pour les détails de ce type de programmation voir du même auteur "PROGRAMMES UTILITAIRES POUR AMSTRAD" (SORACOM) pages 114 à 119.

VITESSE DU GRAPHISME

Dans le BASIC du CPC les positionnements verticaux (ou ordonnées Y) vont de 0 à 400, mais en fait il n'y a que 200 lignes d'écran. Il est donc inutile de progresser en STEP 1 en Y ; en STEP 2 c'est deux fois plus rapide (ou moins lent) et *le résultat est le même*.

Exemple pour tracer un carré rouge au milieu de l'écran

```
CLS FOR Y = 100 TO 300 STEP 2 PLOT 200, Y, 3 DRAWR 200, 0:NEXT
```

Durée 0, 9 seconde, en STEP 1 c'est 1, 8 seconde

Pour des tracés non rectilignes, le manuel conseille de juxtaposer des points par des PLOT, il y a de quoi s'endormir devant l'écran... Non, faites des DRAW (ou DRAWR) suffisamment courts pour que l'on ne remarque pas que cette courbe est une ligne brisée. Non seulement, on est sûr de ne pas avoir un tracé en pointillés en MODE 2, mais le gain de vitesse atteint alors facilement un facteur 10. Exemple, pour dessiner un grand cercle impeccable en 15 dixièmes de seconde, c'est une suite de 36 cordes au pas de 10°.

```
R = 190 X = 320 Y = 200 MODE 2
DEG PLOT X+R, Y, 1
FOR A% = 0 TO 360 STEP 10
DRAW R*COS(A%) + X, R*SIN(A%) + Y
NEXT
```



CONCLUSION

Dix octets par ci, deux millisecondes par là, mais des centaines de fois par une **programmation plus rationnelle** et l'on gagne des kilo-octets et de longues secondes sur un même programme. Il m'arrive souvent de présenter tel ou tel logiciel à des gens qui refusent de croire qu'il est 100 % BASIC, à cause de la rapidité. Obligé de leur faire un BREAK puis un LIST pour les convaincre !

Certes, on n'égalisera pas la vitesse d'un programme en assembleur (surtout en graphisme), mais ces quelques bonnes habitudes sont plus faciles à assimiler et à mettre en œuvre que ce langage.

UN AMPLIFICATEUR DE SON POUR VOTRE CPC

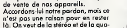
TOUT SUR SOUND

7

Bernard POISOT

Effets stéréophoniques, large gamme de fréquences, composition et maîtrise quasi totale du son, trois voies simultanées, instructions BASIC... Est-ce le début de la campagne électorale ?

Non, ce sont les commentaires flatteurs au sujet des possibilités sonores des CPC, que votre serviteur est l'un des premiers à proclamer et qui vous mettent l'eau à la bouche (ne serait-ce pas plutôt la puce à l'oreille ?). Mais il y a une importante limite à toutes ces belles promesses. La restitution du son se fait par l'intermédiaire d'un minuscule haut-parleur, logé dans le boîtier de l'ordinateur. Cet équipement ridicule n'a qu'une excuse, celle du bas prix de vente de nos appareils.
Accordons-lui notre pardon, mais ce n'est pas une raison pour en rester là. On veut de la stéréo et de la qualité sonore, et on en aura !



LA STEREOPHONIE

C'est le fait que nous possédions deux oreilles qui entendent "chacune de leur côté". Les bruits venant de droite sont perçus par l'oreille droite et bien entendu ceux de gauche par l'oreille gauche. Un son émis devant nous arrivera aux deux oreilles. Cela nous permet de situer le lieu d'émission d'un bruit même sans le voir et d'apprécier le relief sonore. Ceci peut se reconstituer à condition de posséder au moins deux sources sonores : une à droite et une à gauche. Cela explique le fait que le CPC puisse émettre des sons sur deux voies distinctes, qu'il suffit d'exploiter séparément.

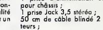
L'AMPLIFICATEUR

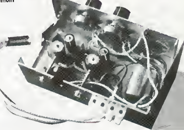
C'est un montage électronique qui a pour but d'amplifier les quelques millivolts disponibles à la prise de sortie stéréo afin de délivrer une puissance permettant une bonne écoute au moyen de haut-parleurs de qualité. L'écoute en stéréophonie implique l'emploi de deux amplificateurs similaires, un pour la voie droite et un pour la voie gauche. Le schéma utilisé n'est pas de haute fidélité, mais il présente divers avantages : une très bonne qualité sonore, une réalisation simple et un coût modique afin de permettre au maximum d'entre vous d'en profiter.
L'implantation des composants a d'ailleurs été étudiée pour que les amstradistes les moins riches s'équipent tout d'abord de la version monophonique, puis il leur suffira, lorsque leurs finances le permettront, d'acquérir et d'y ajouter la seconde partie sans modifier le travail préalable.



NOMENCLATURE

Eléments communs aux deux versions :
Environnement :
1 boîtier métal d'au moins 10 cm de large, 4 de haut et 7 de profondeur ;
1 interrupteur miniature ;
1 diode LED (D1) et son support ;
1 résistance de 470 ohm ;
1 passe-fil d'un centimètre de diamètre environ ;
1 prise d'alimentation électrique pour châssis ;
1 prise Jack 3,5 stéréo ;
50 cm de câble blindé 2 conducteurs ;

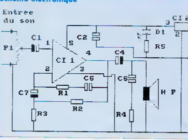




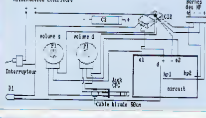
50 cm de petit fil de câblage ;
1 m de fil électrique 2 conducteurs repérés de section 0,75 ou 1 mm² ;
1 plaque de VEROBOARD à bandes cuivrées au pas de 2,54 mm, de 10 x 5 cm ;
1 barrette de raccordement électrique (domino) à 4 plots ;
4 vis, écrous et rondelles diamètre 3 mm ;



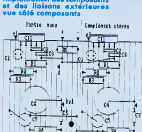
Schéma électronique



Environnement du circuit amplificateur



Implantation des composants et des liaisons extérieures vue côté composants



Alimentation :

Si vous possédez une alimentation à régulation électronique capable de délivrer une tension entre 8 et 18 V et une intensité d'au moins 400 mA (version mono) ou 800 mA (stéréo), vous êtes dispensé d'acheter les composants de l'alimentation que voici :

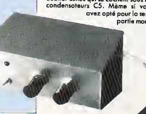
1 condensateur 1000 micro/25 V (C3) ;
1 bloc d'alimentation secteur 9 ou 12 V 500 mA (mono) ou 1000 mA (stéréo) ;
1 circuit intégré 7812 (CI 2)

Amplificateur

Cette liste est à acheter en double pour la version en stéréo :
1 haut-parleur (autoradio, enceinte ou autre) de 6 à 30 W ;
1 potentiomètre de 10 à 47 k logarithmique (P1) ;
1 condensateur radial 10 micro/25V (C1) ;
3 condensateurs 100 nano/63 V (C2, C5 et C6) ;
1 condensateur 1000 micro/16 V (C4) ;
1 condensateur 100 micro/16 V (C7) ;
1 résistance de 47 ohms (jaune, violet, noir) (R1) ;
1 résistance de 150 ohms (marron, vert, marron) (R2) ;
1 résistance de 2,7 ohms (rouge, violet, or) (R3) ;
1 résistance de 1 ohm, 1 watt (marron, noir, noir) (R4) ;
1 circuit intégré TDA 2003 verticol.

MONTAGE

Commencer par effectuer les coupures sur les pistes de cuivre à l'aide d'un cutter. Elles sont représentées par une interruption franche du trait sur le dessin d'implantation. Ne pas oublier celles qui se cachent sous les condensateurs C5. Même si vous avez opté pour la seule partie mono,







il est plus facile de procéder aux préparations lorsque la plaque est dégagée de tout composant. Cela vous facilitera le travail au moment de l'extension stéréo. Il y a 8 coupures. En numérotant les bandes de cuivre de gauche à droite sur le dessin d'implantation (attention car sur ce dessin, elles sont SOUS la plaque) elles se situent sur les bandes 2, 3, 11 (2 interruptions), 1à, 17, 18, 25.

Procéder ensuite au perçage du trou de fixation, piste 13, de 4mm. Repérez les composants dont les pattes ont un diamètre supérieur aux trous de la plaque. Il sera nécessaire d'agrandir les trous où ils se montent.
Souder les fils de liaison entre pistes, en leur laissant leur gaine isolante.
A ce stade, il convient de différencier les deux options. L'amplificateur mono n'utilise que la partie de la plaque qui est à gauche du trait d'axe, et le stéréo toute la plaque et la description suivante se fera en double.
Voici la partie soudure, chacune ne doit pas durer plus de 2 secondes. R1, R2, R3, R5. Pas de sens à respecter pour les résistances.
R4 doit être surélevée d'un cm et ne doit pas toucher C7. En version stéréo, il faut effectuer un pont entre les pistes 18 et 19 côté soudure à l'aide de la patte de R4.
Les condensateurs se montent verticalement. Les plus volumineux ont un sens, l'une des deux pattes est repérée par une inscription. Le sens de branchement est indiqué sur le dessin.
Le circuit intégré CI1 ne doit pas être soudé à ras de la plaque mais par l'extrémité de ses pattes.
La soudure des composants est terminée. Positionnez la plaque dans le boîtier pour organiser l'implantation générale. La plaque reposera sur une rondelle plastique haute de 3 mm et tenue par la vis de fixation pour éviter tout contact avec le métal. CI1 et CI2 (si vous l'utilisez) devront être plaqués contre le boîtier afin d'évacuer la chaleur. CI2 sera fixé par la vis qui tient la barrette de raccordement des haut-parleurs. Le condensateur C3 qui lui est associé sera simplement collé sur le fond du boîtier à l'aide d'adhésif double face et relié par fils souples à CI2. Lorsque la place de chaque élément est définie, repérez la position des trous, videz le boîtier et percez-le. Ne pas oublier le trou de passage des fils.
Remettez tous les éléments sans les fixer définitivement. Cela vous permet de mesurer la longueur des fils nécessaires. Il est préférable de leur donner quelques centimètres de plus pour permettre un démontage aisé du circuit. Utilisez diverses couleurs pour différencier les fonctions par exemple noir pour le moins, blanc pour la diode. Le petit fil sert aux liaisons de D1, C3, P1, le fil plus gros est utilisé pour le reste.
Placez le passe-fil dans le trou de sortie des fils. Procédez aux liaisons. Tous les fils rejoignant le moins du montage doivent arriver au même endroit. Ne laissez pas de grandes longueurs de fil dénudé, cela risque de provoquer des courts-circuits. La plus grande patte de la diode D1 se relie au +. Fixez tous les éléments. Si la vis de fixation de la plaque dépasse sous le boîtier, rehoussez-le avec des pastilles de feutre.
L'arrivée du son se fait par deux conducteurs entourés d'une "tresse" qu'il faudra relier au point d'arrivée des fils moins d'une part et à la grande longuette de la prise Jack d'autre part.
Les deux autres conducteurs se souderont soit au même potentiomètre (patte droite en vue de face), pour l'ampli mono, soit un à chaque potentiomètre en version stéréo.

PRECAUTIONS

Veillez à l'absence de courts-circuits, à la bonne qualité des soudures, à bien immobiliser tous les composants et à ce que la polarité de l'alimentation soit correcte. La plaque ne doit pas toucher le boîtier. En cas de difficulté, un montage aussi simple, s'il a été fabriqué avec soin, sera très facile à dépanner par votre revendeur de composants (apportez-lui le schéma électronique). Notez que le champ magnétique engendré par un haut-parleur est l'ennemi des tubes télé, des cassettes et des disquettes. IL CONVIENT DE LES LAISSER A UNE DISTANCE D'AU MOINS 1 M.

MISE EN MARCHE

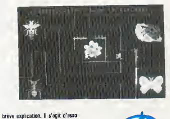
Aucun réglage n'est nécessaire. La diode doit s'allumer dès la mise sous tension et le fonctionnement doit être immédiat. Il est possible que les composants et le boîtier chauffent légèrement en cours de fonctionnement.



SAC A DOS AMSTRAD

Carraz Editions





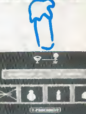
Par rapport au potentiel de logiciels existant sur le marché de l'Amstrad, nous constatons bien souvent que le secteur éducatif n'est pas très développé, quoique heureusement, cette affirmation tende petit à petit à moins se vérifier. Par contre, les petites têtes blondes de maternelle ou même de CP représentent tout un public que nous pouvons qualifier de défavorisé.
C'est pourquoi nous tenons à saluer cette initiative des Editions Carraz qui, de plus, profitent de la fin d'année toute proche pour mettre en place une opération très intéressante. Vous avez en effet la possibilité de vous procurer une disquette regroupant cinq programmes éducatifs et pour compléter ce projet très séduisant, cette disquette est livrée dans un petit sac à dos adorable matelassé tout en coton...
Original, n'est-ce pas ? Avant de décrire chaque programme, sachez que vous pourrez vous procurer le tout pour 290 F TTC.
Le premier programme s'intitule Labyrinthe et s'adresse plus particulièrement aux petits de maternelle. Grâce à un labyrinthe très simple et de beaux dessins évocateurs, l'enfant doit réaliser l'ordre demandé en haut de l'écran : cherche la poule, cherche la fleur...
Le programme suivant s'adresse à toutes les sections de la maternelle et est conçu de telle façon que l'enfant puisse l'utiliser seul après une brève explication. Il s'agit d'associer des objets ou animaux par paire en associant un objet à un autre présenté à l'écran.



C'est alors que nous atteignons l'A.B.C. des petits permettant de travailler sur des mots soit en majuscules, soit en minuscules. De plus, chaque mot à recopier est accompagné d'un dessin le représentant. Enfin, lorsque l'enfant a beaucoup d'entraînement, vous avez la possibilité de rajouter de nouveaux mots.
L'avant-dernier programme propose un jeu qui pourra également satisfaire l'adulte puisqu'il s'agit d'un photo-mémory avec affichage plus ou moins long des dessins (6 par écran).



Pour terminer, il vous est proposé un programme s'intitulant l'Intrus. Après test effectif auprès de petits de la maternelle, ce programme apparaît comme étant le plus difficile de la disquette, il s'agit de cocher dans une famille de quatre images celle qui n'appartient pas à la même famille.
Nous vous recommandons vivement ce sac à dos original avec ses programmes hauts en couleurs, bien dessinés, avec une agréable musique d'introduction avant chaque programme et surtout adaptés à l'enfant.
CARRAZ EDITIONS
Tél. 78 94 10.31.



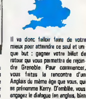
VISA POUR HYDE PARK

Coktel Vision





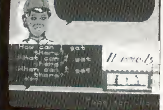
Que diriez-vous de proposer à vos chères petites têtes blondes, qui effectuent cette année leur grande entrée en classe de 6ᵉ, de faire leurs premiers pas en anglais en vivant une aventure ?
En voulant connaître le principe du programme, vous êtes immédiatement pris au jeu en étant propulsé dans une ville d'Angleterre avec une nouvelle identité : vous vous appelez désormais Marc Deray, vous avez 12 ans et habitez Grenoble. Vous connaissez peu de chose des grandes villes et vous vous retrouvez en pays inconnu, comprenant à peine l'anglais et, de surcroît, sans billet d'avion pour retourner en France !



Il va donc falloir faire de votre mieux pour atteindre ce seul et unique but : gagner votre billet de retour qui vous permettra de rejoindre Grenoble. Pour commencer, vous faites la rencontre d'un Anglais du même âge que vous, qui se prénomme Kerry. D'emblée, vous engagez le dialogue (en anglais, bien sûr !) de la manière suivante : votre interlocuteur s'exprime dans une bulle ovale ou ronde et vous, de votre côté, pouvez choisir une expression dans une bulle carrée. Si le choix s'avère difficile ou si vous séchez complètement, vous avez la possibilité de faire appel à une aide qui vous renseigne sur la façon de vous exprimer au moment précis de l'aventure où vous vous trouvez. Vous retournez ensuite là où vous aviez fait appel à l'aide et pouvez donner la bonne réponse.



Kerry vous emmène alors chez lui afin de vous présenter ses parents, ainsi que l'intérieur de sa maison. Bien entendu, les parents de Kerry veulent tout savoir de vous et vous demande votre nom, votre âge, d'où vous venez... Si vous n'êtes pas encore très expérimenté en anglais, une légende en français vous indique ce que l'on vous demande (petit handicap pour la progression car l'enfant n'aura-t-il pas un peu tendance à regarder ce qui est en français ?).
C'est alors que vous recevez un message vous indiquant que vous devez vous rendre à Londres... Grande épopée, quand on sait que cela sous-entend de prendre le train et donc d'acheter un billet pour le voyage ! Heureusement que Kerry accepte de vous accompagner !
Une fois arrivé au point de rendez-vous à Londres, vous devez effectuer des achats afin de pouvoir obtenir des renseignements supplémentaires, ce qui vous oblige à communiquer avec d'autres personnes... Ce petit séjour bénéfique va finalement se terminer dans Hyde Park où vous pourrez récupérer (enfin !) votre cher billet de retour pour la France...
Avec Visa pour Hyde Park, vous devez réagir face à des situations précises et devez communiquer avec des personnages ; c'est pourquoi l'aide qui est fournie ne se présente pas comme une explication rigoureuse d'un point de grammaire, mais plutôt comme une remarque vous permettant de trouver la bonne réponse qui vous échappait...
Tout ce cheminement, qui est très progressif, est illustré par des écrans d'allure vraiment très anglo-saxonne, avec un choix de couleurs agréables.
Le fait d'allier l'aventure avec l'apprentissage d'une langue rend ce logiciel attirant. De plus, les notions abordées sont simples et clairement expliquées. Enfin, pour compléter le tout, une cassette audio est livrée avec le logiciel, ce qui vous permet d'entendre effectivement tous les points d'aide... Heureuse initiative vous permettant de vous entraîner sur la prononciation.
Après ce petit voyage rapide, je ne vous dirai qu'une seule chose : "How do you do ?"...



SUR TOUT CE QUI BOUGE

Quand votre QUICKSHOT 2 est branché, nous dit Hugues MONGES, de Chatelleroult, votre CPC affiche un "Z" dès que vous l'allumez ou à chaque RESET. De ce fait, certains programmes (SPITFIRE 40 par exemple) se plantent.
Pour modifier ce comportement étrange, il vous suffit de démonter le socle du joystick, de débrancher le fil rouge de l'AUTOFIRE et de remonter le socle en prenant soin de ne pas couper les fils du bouton FIRE qui se trouvent dans le ressort. Plus simple, tu meurs !

TOP CHRONO

Philippe LEDUC, de Rouen, nous indique la façon de remettre à zéro le compteur TIME - voir LES CHRONOMETRES du CPC N° 22 page 82.
Il s'agit de POKER à zéro les adresses &b187, &b188, &b189 et &b18A. En langage d'assemblage, cela donne :
LD 0E,0
LD HL,0
CALL &bd10
&bd10 est le vecteur qui permet à l'interpréteur de positionner le TIMER à une valeur choisie.

MBMOTSB

Quand, dans le traitement de texte AMSWORD, le fichier que l'on veut charger est erroné ou absent de la disquette, le message d'erreur XXX NOT FOUND s'affiche suivi de READY. Il faut alors éditer le catalogue pour s'y retrouver. Jacques VOLBRECHT, de Château-Thierry vous propose quelques lignes BASIC pour un affichage automatique de catalogue.
Il s'agit, en premier lieu, de charger le fichier MBMOTSB.BAS par LOAD et d'étudier ensuite la ligne 1750 que l'on va transformer ainsi :

```
1750 ON ERROR GOTO 3000   MEMORY mh   OPENIN aS
     CALL c+45-3*a CLOSEIN MEMORY m1   RETURN
```

Nous entrons alors en 3000 les quatre lignes suivantes :

```
3000 MODE 2 CAT aS=" "
3010 PRINT PRINT"Appuyez sur une touche"
3020 IF INKEY$=" "THEN 3020
3030 MODE 1 RESUME 1700
```

Une petite sauvegarde de l'ensemble par SAVE"MBMOTSB" et le tour est joué. L'astuce supplémentaire est, quand on a oublié le nom d'un fichier, d'entrer un nom bidon, ce qui provoquera l'affichage automatique du catalogue. Malin, non ?

CODES MACHINE

Nous voyons dans le courrier que nous recevons, qu'un certain nombre de lecteurs possèdent une imprimante FASTEXT-80 de SMITH-CORONA ce qui leur pose un certain nombre de problèmes dans l'adaptation des codes imprimante aux normes courantes EPSON. D'autre part, nous avions publié dans CPC HS n° 5, un programme, MACHINE A ECRIRE, qui permet d'utiliser votre imprimante en direct à partir du clavier. Gérard JAMBIM, de Hérouville St Clair, nous offre une adaptation de ce programme à la FASTEXT-80. Quelques modifications suffisent.

```
460 WINDOW#4,1,80,1,1:LOCATE#4,8,1
    PRINT"M A C H I N E  A  E C R I R E"
650 CLS#2:PRINT#2,"A=NORMAL B=S.I C=ELITE
    1=GRAS 2=SOULIGNE 3=GRAS+SOULIGNE"
670 CLS#3:PRINT#3,"D=A1    E=A2    F=A3
    G=B1    H=B2    I=B3    J=C1    K=C2
    L=C3"
760 SOUL=0:SL=0:B$=INKEY$:B$=UPPER$(B$)
    IF B$="" THEN 760
770 B=ASC(B$)-64:IF B<1 OR B>12 THEN 760
780 CLS#2:PRINT#2,"A=NORMAL B=S.I C=ELITE
    1=GRAS 2=SOULIGNE 3=GRAS+SOULIGNE"
790 CLS#3:PRINT#3,"D=A1    E=A2    F=A3
    G=B1    H=B2    I=B3    J=C1    K=C2
    L=C3"
800 ON B GOTO 840,880,920,960,1000,1040
    1080,1090,1110,1120,1130,1135,1140
840 PRINT#8,CHR$(27);"P";:CLS#2:PRINT#2,I$;
    " ECRITURE NORMALE ";I$;:CLS#3:
    PRINT#3,"<TAB>=CHOIX D ECRITURE":
    GOTO 1140
880 PRINT#8,CHR$(15);:CLS#2:PRINT#2,I$;
    "  B = MODE SHIFT.IN  ";I$;:CLS#3:
    PRINT#3,"<TAB> = CHOIX D'ECRITURE":
    GOTO 1140
920 PRINT#8,CHR$(27);"M";:CLS#2:PRINT#2,I$;
    "  C = MODE CONDENSE ";I$;:CLS#3:PRINT#3,
    "<TAB> = NOUVEAU CHOIX":GOTO 1140
960 PRINT#8,CHR$(27);"P";CHR$(27);CHR$(87);
    CHR$(1);:CLS#2:PRINT#2,I$;" D = MODE
    NORMAL - GRAS ";I$:MEM=X:X=X/2:XX=1:
    GRAS=1:CLS#3:GOTO 1140
1000 PRINT#8,CHR$(27);CHR$(45);CHR$(1);
    CHR$(27);"P";:CLS#2:PRINT#2,I$;"E =
    MODE NORMAL - SOULIGNE ";I$;:SOUL=1:
    CLS#3:GOTO 1140
1040 PRINT#8,CHR$(27);CHR$(45);CHR$(1);
    CHR$(27);"P";CHR$(27);CHR$(87);CHR$(1);:
    CLS#2:PRINT#2,I$;" F = MODE NORMAL
    - GRAS - SOULIGNE ";I$;:MEM=X:X=X/2:
    XX=1:GRAS=1:SOUL=2:CLS#3:GOTO 1140
1080 PRINT#8,CHR$(15);CHR$(27);CHR$(87);
    CHR$(1);:CLS#2:PRINT#2,I$;" G = MODE
    SHIFT.IN - GRAS ";I$;:MEM=X:X=X/2:
    XX=1:GRAS=1:CLS#3:GOTO 1140
1090 PRINT#8,CHR$(15);CHR$(27);CHR$(45);
    CHR$(1);:CLS#2:PRINT#2,I$;" H = MODE
```

■ Pierre TACONNET

```
      SHIFT.IN - SOULIGNE ";I$;:SOUL=1:
      CLS#3:GOTO 1140
1110 PRINT#8,CHR$(15);CHR$(27);CHR$(45);
      CHR$(1);CHR$(27);CHR$(87);CHR$(1);:
      CLS#2:PRINT#2,I$:" I = MODE SHIFT.IN
      - GRAS - SOULIGNE ";I$;:MEM=X:X=X/2:
      XX=1:GRAS=1:SOUL=2:CLS#3:GOTO 1140
1120 PRINT#8,CHR$(27);"M";CHR$(27);CHR$(87)
      CHR$(1);:CLS#2:PRINT#2,I$:" J = MODE
      ELITE - GRAS ";I$::MEM=X:X=X/2:XX=1:
      GRAS=1:CLS#3:GOTO 1140
1130 PRINT#8,CHR$(27);CHR$(45);CHR$(1);
      CHR$(27);"M";:CLS#2:PRINT#2,I$:" K =
      MODE ELITE - SOULIGNE ";I$;:SOUL=1:
      CLS#3:GOTO 1140
1135 PRINT#8,CHR$(27);CHR$(45);CHR$(1);
      CHR$(27);"M";CHR$(27);CHR$(87);
      CHR$(1);:CLS#2:PRINT#2,I$:" L = MODE
      ELITE - GRAS - SOULIGNE ";I$;:MEM=X:
      X=X/2:XX=1:GRAS=1:SOUL=2:CLS#3:GOTO 1140
1650 PRINT#IMF,CHR$(18);CHR$(27);CHR$(45);
      CHR$(0),CHR$(27);"A";CHR$(7);CHR$(27);
      CHR$(82);CHR$(3);CHR$(27);CHR$(87);
      CHR$(0);
1730 IF ASC(A$)>47 AND ASC(A$)<58 THEN 1790
1740 IF ASC(A$)>63 AND ASC(A$)<91 THEN 1780
1750 IF ASC(A$)>96 AND ASC(A$)<126 THEN 1800
1780 ON ASC(A$)-63 GOTO 2100,1840,1850,1860,
      1870,1880,1890,1900,1910,1920,1930,1940,
      1950,1960,1970,1980,1990,2000,2010,2020,
      2030,2040,2050,2060,2070,2080,2090
1800 ON ASC(A$)-96 GOTO 2260,2270,2280,2290,
      2300,2310,23120,2330,2340,2350,2360,2370
      2380,2390,2400,2410,2420,2430,2440,2450,
      2460,2470,2480,2490,2500,2510,2520,2530,
      2540
2100 A$=AA$:RETURN
2520 A$=EE2$:RETURN
2530 A$=UU2$:RETURN
2540 A$=EEE2$:RETURN
4311 SYMBOL 245,&60,&10,&3c,&66,&7e,&60,&3c
4312 EEE2$=CHR$(245)
4313 SYMBOL 250,&30,&8,&66,&66,&66,&66,&3e
4314 UU2$=CHR$(250)
4315 SYMBOL 247,&6,&8,&3c,&66,&7e,&60,&3c
4316 EE2$=CHR$(247)
4317 SYMBOL 248,&60,&10,&78,&c,&7c,&cc,76
4318 AA$=CHR$(248)
```

Et voilà comment CPC + FASTEXT-80 deviennent machine à écrire ! Deux petites réserves, taut de même, sant à émettre le ''ç'' n'apparaît pas à l'écran en made gras et il reste impassible de corriger les fautes de frappe par retour en arrière

LOGO CONTRE LOGO

Yves CABON, de Bischwiller, trouve que son confrere en LOGO, Jean-Philippe UMBERT, de Belfort, a pris des chemins tortueux, dans LE LIEVRE ET LE LOGO (Trucs et Astuces n°25) C'est donc un derby entre le Bas-Rhin et le Territaire de Belfort Secande manche, caup de gong

1 - Il suffit d'un lecteur de disquettes pour transferer des fichiers à l'aide de PIP COM

2 - Si on ne place pas sur la disquette le fichier SETKEYS COM, l'ordre SETKEYS KEYS DRL du fichier PROFILE SUB ne sera pas exécuté

3 - Améliaration 1 : ajouter sur vatre disquette (avec PIP COM !) les fichiers LANGUAGE COM et PALETTE COM LANGUAGE COM vaus permet de dispaser de tous les alphabets natianaux Au lancement de CP/M, c'est l'alphabet américain qui est pris par défaut LANGUAGE 0 Pour obtenir les caractères français il faut utiliser le LANGUAGE 1 PALETTE COM vous permet de modifier les cauleurs de l'écran Si vous n'aimez pas le blanc sur fond bleu, chaisissez vas cauleurs en vous aidant du tableau qui se trauve, dans le manuel du 6128, au chapitre 5 Page 25

3 - Amélioratan 2 plutôt que d'entrer en mode direct vos ardres dans PROFILE SUB, il est préférable de créer un petit programme BASIC qui a l'avantage de pauvoir être sauvegardé

```
10 OPENOUT ''PROFILE SUB''
20 PRINT#9,''LANGUAGE 1'' 'Alphabet français
30 PRINT#9,''SETKEYS KEYS DRL'' 'config clavier
40 PRINT#9,''PALETTE 35,0'' 'cyan sur noir
50 PRINT#9,''LOGO3'' 'lance LOGO
60 CLOSEOUT
```

Si l'an y regarde d'un peu près, seule l'absence de SETKEYS.COM semblait poser problème La balle est toutefais dans le camp de Belfort. Le public se passionne et attend la traisième manche !

DES GOUTS ET DES COULEURS

Le programme FONCTIONS EN COULEURS de CPC n° 24 n'était pas daté de son catalague de fonctions que vous trouverez dans CPC n° 27 Mais ce n'est pas pour cela que Richard CARLIER, qui ne naus donne pas son adresse, nous écrit Il prapose une astuce paur sauvegarder les images créées et une façan originale de les charger.

Paur la sauvegarde

```
125 SAVE nom$+''.img'',b,&c000,&4000
1905 LOCATE 1,15 PRINT''Entrez le nom de l'image à
'' INPUT''sauvegarder (8 caractàres max) '';nom$
```

Paur le chargement

```
10 IDIR,''* IMG''
20 LOCATE 10,20·INPUT''Nom de l'image: '',nom$
30 'placer ici le choix des cauleurs, lignes 3030-3060 et 3080
40 PAPER 0.CLS BORDER 0
50 LOA0 nom$+''.img''
100 placer éventuellement ici le sous-programme de variatians de couleurs, lignes 5000-5540
200 OUT &bc00,1.FOR X=1 TO 40
210 FOR T=1 TO 50 NEXT T
220 OUT &bd00,X NEXT X
300 FOR P=1 TO 2000.NEXT P RUN ou END
```

Voilà un petit camplément qui devrait susciter l'intérêt.

UTILITAIRE

Bourse 2000

Avec les privatisations nouvelles, les petits actionnaires vont devenir légions. Cette entrée en matière donnera peut-être à tout un chacun l'envie de se lancer plus à fond dans l'aventure de la bourse. Après de multiples hésitations, on songe que l'ordinateur, si apte à manier les chiffres, pourrait être le partenaire idéal de l'apprenti boursier. Bourse 2000 se propose d'être le compagnon de votre aventure. Car aventure, il y a. N'oublions pas que la Bourse présente quelques risques pour celui qui le côtoie. L'utilitaire présenté ici vous aidera justement à réduire cette part de risque. Mais voyons tout d'abord le fonctionnement du logiciel et ses diverses options. Cette partie est relativement restreinte puisque la manipulation est facile et l'on se sert très peu des touches du clavier (sauf, bien sûr, lors de l'entrée des cotes). D'ailleurs, le manuel de quarante pages ne consacre que la moitié de son contenu au mode d'emploi du logiciel.

Au départ, on trouve un menu de quatre options : gestion, graphiques, affichage, calculs.

Gestion : il s'agit de la partie "manipulation" des données. C'est à dire l'entrée des valeurs et de leurs cours. La capacité d'un fichier est de quarante titres, mais il est bien sûr possible d'utiliser les cotations sur trois ans à partir de l'année de référence. Lorsque ces trois années auront été complétées, il suffira d'effacer l'année la plus ancienne pour récupérer de la place. L'entrée des cotations se fait par l'intermédiaire de deux menus. Le premier permet de considérer des cours non cotés, ainsi que les années ne possédant pas 53 semaines de cotations (53 est en effet le nombre maximum de semaines admises). Le deuxième menu accessible par le premier "automatise" l'entrée des données. Les autres options sont assez parlantes : lecture et modification des cours, suppression d'une valeur, suppression d'une année de cours, impression des valeurs (avec indication de la première et de la dernière cotation) et, enfin, choix du fichier de travail qui vous laisse l'option à utiliser pour créer un nouveau fichier.

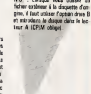
Le mode graphique offre l'avantage d'une lisibilité plus grande. Les variations des cotes sont immédiatement visualisées sur les trois années avec plus de deux axes en pointillés permettant des comparaisons entre deux graphiques. Diverses caractéristiques sont affichées telles que pourcentage maximum et minimum, cotation maximale ou minimale, ainsi que la croissance par année. La moyenne mobile permet d'éliminer les variations brusques, on obtient une courbe "lissée". Un appui sur la touche I et une hard-copy couche le graphe sur papier.

Le mode "calculs" donne les hausses et baisses moyennes, le taux de croissance, la genèse (c'est-à-dire l'état d'une valeur sur une période donnée) et enfin la situation du fichier qui récapitule dans un tableau les tendances de chaque valeur.

La deuxième partie du manuel est à mon avis la plus intéressante puisque l'on vous explique les divers procédés de prévisions des hausses et baisses. On s'aperçoit alors que les mouvements des cotations peuvent être plus modélisés à l'aide des courbes et des graphiques points et figures. Vous saurez tout sur les moyennes mobiles, la vitesse, l'élasticité, le signal, la méthode des canaux. A vous donc de choisir parmi les méthodes ou de les cumuler afin de réduire au maximum le risque.

Bourse 2000, de par sa simplicité d'utilisation, (les touches les plus employées sont les flèches du curseur et la barre d'espace), ses possibilités graphiques et son module d'impression saura vous guider dans la jungle boursière.

N.B. : Lorsque vous utilisez un fichier extérieur à la disquette d'origine, il faut utiliser l'option drive B et introduire le disque dans le lecteur A (CP/M oblige).



COURS D'UNE VALEUR

BOUYGUES 1986

semaine	0	1	2	3	4	5
0		851.00	1241.00	1310.00	1424.00	1327.00
1	1119.00	837.00	1060.00	1314.00	1350.00	1270.00
2	1019.00	1045.00	1140.00	1330.00	1270.00	1270.00
3	1030.00	1065.00	1070.00	1450.00	1360.00	
4	1025.00	1068.00	1105.00	1445.00	1345.00	
5	940.00	1105.00	1145.00	1445.00	1255.00	
6	1040.00	1200.00	1172.00	1423.00	1282.00	
7	1075.00	1234.00	1205.00	1325.00	1290.00	
8	360.00	1176.00	1204.00	1270.00	1279.00	
9	953.00	1245.00	1220.00	1340.00	1279.00	

ESAT Software 1987

UTILITAIRE



GENEDIT

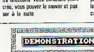
Genedit est un logiciel bicéphale, c'est-à-dire qu'il contient en fait deux programmes indépendants et complémentaires.
Le premier, Génécar, est un générateur de caractères pouvant affecter 255 caractères au maximum. Rappelons que l'Amstrad possède, dès sa mise en marche, un jeu de caractères standards. Ceux-ci peuvent être modifiés grâce aux instructions SYMBOL AFTER et SYMBOL. En général, il est plus pratique d'utiliser un programme de redéfinition plutôt que d'entrer une série de chiffres correspondant à l'alphabet gothique dont la réalisation a demandé plusieurs semaines de dur labeur ! Ce genre de programme se doit d'être clair et simple d'utilisation.

Génécar débute par un menu de sept options couplé à un tableau complet de 255 caractères disponibles. La première option, charger un caractère, vous invite à choisir le symbole sur lequel vous voulez travailler. Après un appui sur COPY, on accède à une matrice agrandie de 8 sur 8 cases. Juste à côté figure les représentations du caractère en mode 2 et en mode 1.
Les flèches du curseur permettent de se déplacer sur la matrice, alors que la touche COPY "allume" ou "éteint" un point. Il est même possible de déplacer le contenu de la matrice entière dans une des quatre directions. Votre caractère étant créé, vous pouvez le sauver et passer à la suite.

Lorsque vous êtes satisfait du résultat, sauvez votre œuvre sous forme d'un fichier .CAR. Ce fichier est utilisable dans tout programme BASIC.
Editcar : second programme de Genedit, il utilise les possibilités de Génécar. En effet, les fichiers CAR déjà créés peuvent être chargés et ce, en cours de création d'écran, car Editcar est un éditeur de pages écran en mode texte ou semi graphique (à l'aide des caractères graphiques). Les caractères affichables sont ceux normalement accessibles au clavier.
Sur l'écran presque vide du CPC, on trouve le curseur qui peut être déplacé avec les flèches. Les touches DEL et CLR sont toujours actives et vous permettent d'effacer des caractères.
Il existe une série de bascules actionnées soit au menu, soit à partir de l'écran d'édition. Parmi ces bascules, on trouve la surimpression, la commutation mode 1/mode 2, la vidéo inverse et la transparence. La surimpression est utilisée pour remédier à un défaut gênant : le décalage de toute une ligne de caractères lors d'un ajout.

Le programme est de type WYSIWYG (What You See Is What You Get : ce que vous voyez est ce que vous obtenez). En effet, l'option 6, impression de l'écran, aura pour résultat une feuille contenant exactement ce qui a été inscrit à l'écran.
Ce programme simple d'emploi et sans prétention vous permettra d'obtenir des pages titres (n'oubliez pas de réserver 17 Ko pour chaque image) très facilement.

0 1 2 3 4 5 6
@ A B C D E F
P Q R S T U V
` a b c d e f
p q r s t u v

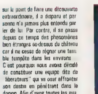


Les possibilités de Génécar associé à Editcar, sont sans précédent:
- Mode 40 ou 80 caractères par ligne sur le même écran
- inversion couleur dans les 2 Modes
- Chargement, pendant la fabrication de l'image, de toute police de caractères crée par vous avec Génécar (α,β,θ,δ,ε - Ha зaaposbe!)
- Programmes très simples d'emploi, très "amicaux"
- Sort sur l'imprimante exactement ce que vous voyez sur cet écran

LOGICIELS

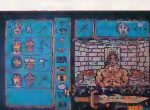
LE MAITRE DES AMES

UBI SOFT
Jeu de rôle/Aventure



En ces temps lointains la légende dit qu'il existait un très grand magicien que nous appelions Khaman Akkad et que tout le monde craignait tant son pouvoir était important et étrange

Mais un beau jour, alors qu'il était sur le point de faire une découverte extraordinaire, il a disparu et personne n'a jamais plus entendu parler de lui Par contre, il se passe depuis ce temps des phenomènes bien étranges au-dessus du château car il ne cesse de régner une terrible tempête dans les environs
C est pourquoi nous avons décidé de constituer une equipe dite de libérateurs ' qui va oser affronter son destin en pénétrant dans le donjon. Afin d'avoir toutes les qualites reunies, ainsi qu'une puissance d armement, l'equipe se constitue des quatre membres suivants , un magicien, un guerrier, un voleur et un elfe... Nous penétrons tous ensemble dans ce donjon de pierres froides et partons à la recherche des premiers indices et des premières rencontres qui s'avèrant parfois malheureuses, comme avec le gros rat, car il n'y a pas d'autre issue que le combat dans ce cas, ou bien alors fructueuses comme avec le cuisinier du château qui, contre quelques écus, nous livre ce message enigmatique . "Méfiez vous du chant des champignons hurleurs..." ce qui nous promet une belle suite !



Ce nouveau logiciel d'Ubi Soft a tous les critères d'une réussite dans la catégorie jeu de rôle grâce à un graphisme flamboyent et à un système d'actions se faisant par des icônes imagées .. La phase de création d'une équipe passe par un tirage au sort des qualités avant le choix des armes et de l'armure Somme toute, une nouveauté qui va faire parler d'elle...

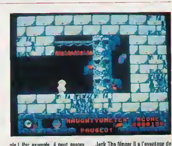
JACK THE NIPPER II

GREMLIN GRAPHICS
Arcade

Jack la Terreur est de retour ! En effet, il y a environ un an, ce "cher petit" sévissait chez lui et dans sa ville en réalisant les pires bêtises qu'il pouvait trouver à commettre , et il en était fier !
A présent, toute la famille est en route pour l'Australie, ce qui est loin de satisfaire Jack Alors, pendant que l'avion survole la jungle, notre petit ami prend sa couche, compte jusqu'à trois, retient sa respiration et hop ! Saute dans le vide Sa couche lui sert de parachute et lui permet d'atterrir sans encombre au milieu de ce paysage inhospitalier... Seulement, il y a un petit problème : papa a suivi et il va falloir éviter de se faire rattraper car, sinon, c'est la fessée assurée !



Dans ce nouvel épisode, Jack a toujours son échelle de détestabilité à faire grandir en faisant le plus grand nombre possible de méchancetés et ce n'est pas ce qui manque dans le dur milieu de la jungle ! Par exemple, il peut assommer ses adversaires à coups de noix de coco ou en utilisant une sarbacane. Il y a également plusieurs objets qui ne sont pas négligeables tels de la graisse, du miel ou un bouclier.. A votre charge de trouver une utilisation originale de tous ces objets !



Jack The Nipper II a l'avantage de présenter une multitude d'écrans avec de nombreuses actions différentes Graphisme et animation sont propres et nets sans être pour autant exceptionnels Mais l'assurance de défoulement est là !

LOGICIEL



Jeu

L'unique, le vrai, le prototype de l'aventurier s'anime devant un tube cathodique en l'occurrence celui du moniteur Amstrad On le reconnaît facilement : sa silhouette altière, son chapeau usé et son fouet Ce héros au sourire si doux s'est déplacé aux Indes. En effet, il se passe des choses étranges dans le village de Malapore. Une pierre sacrée a été volée le Sankara. Celle-ci protégeait le village des agissements du Maharadjah malfaisant et du grand prêtre Mola-Ram (mémoire molle ?) Ce qui devait arriver arriva ! Les enfants du village ont été mis en esclavage sous la surveillance des gardes Thuggee, ils fouillent les mines à la recherche des pierres précieuses.

Bien entendu, en tant que héros dévoué à la cause du bien, vous ne pouvez laisser passer cette manifestation du mal. Vous voilà donc en grande tenue, à errer dans les coins et les recoins de la mine. Le joystick en main vous dirigerez votre personnage (Indy) de tous côtés en évitant de tomber des plates-formes Il est préférable d'utiliser les échelles pour monter et descendre Le fouet, outil universel, va être utilisé pour étourdir les gardes Thuggee, se lancer d'une galerie à l'autre et enfin délivrer les enfants.

Il y a neuf enfants à délivrer pour changer de niveau Attention, le moindre contact avec les gardes ou avec les boules de feu lancées par l'immonde Mola-Ram vous fait perdre une vie

Le niveau 2 vous propose un voyage à bord d'un wagonnet avec moult rebondissements et chauves-souris. Puis, si vous y parvenez, le niveau 3 vous entraîne dans le "Temple Maudit" à la recherche de la pierre sacrée. Si vos articulations tiennent jusque-là vous avez gagné

N'espérez pourtant pas retrouver l'ambiance du film : les graphismes et l'animation sont vraiment moyens





LE CRÉPUSCULE DU NAJA

Les "aventures du Ka"

Jeu

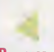
Ce jeu d'action vous entraîne dans les moiteurs d'une jungle asiatique Ka est le sigle du nom d'un agent ultra-secret Karl Adrix. Votre mission est également ultra-secrète donc pas d'ordre spécifique sinon celui de faire très attention où l'on pose les pieds. Puisque vous n'êtes que le compagnon du héros (vous savez celui qui pose avec les belles dames, qui se nourrit exclusivement de caviar et de champagne) on vous a laissé le soin de diriger KA au joystick Pas de problème, vous êtes justement un as du manche à balai Un petit tableau vous donne une vue d'ensemble des ennemis potentiels et des armes mises à votre disposition Le programme démarre très fort puisque dès le premier tableau un monstroïde arachnide se jette férocement sur vous Ceci a pour effet de "pomper" votre énergie. En effectuant de gracieux bonds au dessus des touffes d'herbes disposées çà et là, vous parviendrez à échapper à l'abominable en traversant un rideau de lianes coulissant. Et hop, un autre tableau, lui aussi plein d'une végétation luxuriante avec lianes, bambous et même des plantes carnivores. Quelques objets seront utiles à votre quête de l'inconnu les haches, les pistolets et les réacteurs. Les deux premiers sont destinés à l'anéantissement

connu les haches, les pistolets et les réacteurs Les deux premiers sont destinés à l'anéantissement des diverses bestioles volantes qui vous agressent perpétuellement. Le réacteur dorsal, lui, permet d'économiser beaucoup de temps et d'énergie en volant au dessus des obstacles tels que têtes de mort, plantes carnivores qui elles aussi vous aspirent de l'énergie De temps à autre on retrouve une tête de Bouddha posé nonchalemment sur le sol Installez-vous dessus et bénéficiez du tout-à-l'énergie qui vous regonflera le moral.

Le graphisme et l'animation sont plutôt réussis mais le scénario reste vague et le style de déplacement rappelle celui de Sorcery.
Une suite est annoncée le talisman d'Dsins



LOGICIELS

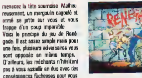


RENEGADE

Jeu

Quelle idée d'avoir accepté ce rendez-vous à cette heure de la nuit! Golda aurait pu attendre pour une fois ! Il faut dire que la perspective d'emprunter le métro de New York à une heure aussi tardive n'était pas pour vous enchanter. Votre prémonition était bonne. Sur le quai de votre station, une bande de loubards vous attend. Vous êtes seul, ils sont six dont deux armés de battes de base-ball. Leurs regards vous persuadent qu'ils ne sont pas là uniquement pour vous demander l'heure. En effet, ils se dirigent vers vous d'un pas décidé à vous marcher dessus. Heureusement, vous n'êtes pas totalement sans défense : vous avez une certaine pratique des arts martiaux. Cela va vous rendre de grands services. Vous vous approchez du premier vilain et vous lui envoyez un coup de poing que Rocky lui-même n'aurait pu éviter. L'affreux s'étale mollement. Mais voilà qu'un autre de ces voyous vous prend à revers : cette manœuvre ne surprend pas votre œil de lynx et d'un pied vengeur, vous menacez la tête sournoise. Malheureusement, un margoulin cagoulé et armé se jette sur vous et vous frappe d'un coup imparable.

Voici le principe du jeu de Renégade. Il est assez simple mais pour une fois, plusieurs adversaires vous sont opposés en même temps. D'ailleurs, les méchants n'hésitent pas à vous assaillir en duo avec des conséquences fâcheuses pour vous. Passons au gros problème du logiciel : le déplacement et l'attaque. Au clavier, il y a 4 touches de déplacement et 3 touches pour l'attaque. En mode joystick, le déplacement s'effectue bien sûr avec le manche, en revanche l'attaque s'effectue toujours au clavier, d'où une certaine angoisse existentielle pour les possesseurs de speedking ou de Magnum et autres Phaser One (comprenne qui pourra). Il faut donc se procurer au plus vite un manche à ventouses pour espérer progresser au long des 3 niveaux. C'est un peu dommage car les décors sont bien jolis.







COBRA

Jeu

Quel est le point commun entre les logiciels Amstrad et les séries télévisées ? La réponse est simple : les adaptations. Cobra fait partie de ces reprises informatiques. On y retrouve l'homme au cigare et au bras laser. Comme de bien entendu, le méchant de service a fait des siennes et la charmante Dominique s'est retrouvée captive en un tour de main. Dominique, est-il besoin de le rappeler, est la fiancée de notre héros. Je suppose que ces quelques lignes vous ont déjà donné la clef du scénario. Cobra, accompagné de la dame de fer (mais non, il ne s'agit pas du premier ministre anglais, mais de l'androïde femelle Armanoïde). Puisque vous connaissez le principe, vous brûlez de découvrir le comment. La stratégie est assez simple : tirer dans le tas.

Voici les faits : votre duo déambule dans un décor dénudé à l'aide d'un unique joystick. Les seuls obstacles (mis à part les pseudo humanoïdes) rencontrés sont les murs d'un labyrinthe. Une petite étoile apparaît parfois dans un coin de l'écran. Elle indique la direction à suivre pour retrouver Dominique et son tortionnaire.

De chaque côté de l'écran de jeu, il y a les représentations graphiques des deux héros. Juste au dessous, une série de 3 petites lampes clignotantes indiquent le niveau d'énergie de chacun des protagonistes. Si toutes les lumières s'éteignent, le jeu s'arrête. Il faut donc éviter tout contact brutal avec les objets volants pas très identifiés qui circulent sur toute la surface du jeu. Les déplacements ne sont pas facilités par une gestion du joystick plutôt bizarre. Un autre détail : on peut très bien rapprocher ou éloigner les personnages l'un de l'autre, on obtient ainsi des faisceaux de tir plus ou moins proches.

Armanoïd peut être activée ou désactivée. Dans ce cas, Cobra continue seul et sans possibilité de tir. Il lui reste heureusement cinq grenades fort utiles. Ce programme s'il présente des dessins assez fins (nous sommes en MODE 1) n'offre pas un scénario enthousiasmant.



TAI PAN

Jeu

1841, dans la ville de Canton, vous et votre fils avez décidé de faire fortune grâce à la contrebande Pour cela, il vous faudra trouver de l'argent pour acheter un bateau, pour engager un équipage et pour équiper le bâtiment En effet, à cette époque le commerce de contrebande par voie maritime est florissant Bien entendu qui dit gros profits dit aussi pirates Vous le comprenez, cette aventure ne va pas être de tout repos Première étape la recherche de l'argent En bas de l'écran, on trouve une série de 4 icônes L'une represente une bourse ouverte vers l'exterieur, elle figure l'action d'acheter, le symbole symétrique représente la vente Une main surmontant une flèche c'est la prise d'un objet Les deux dernières icônes sont utilisees pour la sauvegarde et le chargement du jeu (sage précaution en effet, les debuts sont parfois penibles). La ville va vite revêtir un aspect monotone pour vous car les décors sont souvent très semblables. On peut trouver l'argent nécessaire à l'achat du bateau dans le restaurant. Il suffit de refuser le repas et l'on vous prêtera 300000 $ (il faudra absolument rembourser cette somme au bout d'un mois)

Il vous est possible d'aller à la banque pour acheter votre bateau Il y a 3 modeles au choix selon vos besoins et vos finances L'équipage peut être soit achete soit enrôle de force Dans ce dernier cas, méfiez vous, ces hommes risquent de ne pas être tres loyaux envers vous En certains lieux on vous invitera a participer à un jeu de hasard, ressemblant a un tiercé Les contrebandiers vous vendront de la marchandise Vous pouvez l accepter mais soyez vigilants si l'on vous trouve en possession de ces objets prohibes vous allez directement en prison Il vous restera à decouvrir la seconde partie comprenant les trajets en bateau entre les îles ou les attaques des navires marchands (car vous pouvez devenir pirate également) Le graphisme ne retient pas vraiment l attention en revanche l'histoire tirée du best seller de Clavell semble être interessante



PRESIDENT

Jeu

Ce programme a pour ambition de vous porter à la tête d une petite nation Vous pouvez choisir l'état initial de votre pays fort, moyen, faible

Evidemment au niveau "faible", la tâche sera plus rude Voici les différents secteurs de votre action les sondages, l'économie, l'industrie, les importations, la santé

Votre but : vous faire réélire après 2 ans de pouvoir Plusieurs tableaux vont défiler : le premier est une représentation graphique du territoire national avec au départ une installation de forage et de stockage ainsi que 2 unités antiériennes C'est bien peu car voici qu'apparaissent les tanks ennemis Ceux ci sont assez nombreux, il va falloir envisager une riposte. Donc choisissez l'option "importer un tank" pour augmenter votre capacité de défense Après cette concession militaire, il faut penser à la population civile Plantez donc un champ de céréales, en général, il s'agit du secteur ou la demande est la plus forte. Chaque "coin" du terrain ne peut être occupé que par un seul objet De plus vous ne pouvez placer d'objets sur les icônes représentant un lac. Les stations de forage et de stockage du pétrole doivent être implantées dans les zones riches (d'où l'intérêt de l'option surcray qui dévoile les ressources du sous-sol) Mais ce n'est pas tout, il faut aussi construire des routes pour les camions citernes

La difficulte vient du fait qu'une seule action est possible il ne faut absolument pas se tromper sur les choix prioritaires Aprés cette première phase, on trouve une série de tableaux concernant la vente ou l'achat d or, les contrats avec les pays étrangers, les importations, l aide à fournir au secteur medical pour enrayer certaines épidémies (cholèra et autres babioles)

Les Nations-Unies rendent également leur verdict sur votre manière de gouverner, vous pouvez être un bon dirigeant ou bien même un tyran.

Si le graphisme n'est pas le point fort du programme, le petit dictateur qui sommeille en chacun de nous pourra pleinement s'exprimer

